## CS 591 S2—Formal Language Theory: Integrating Experimentation and Proof—Fall 2018

# Problem Set 4

### Model Answers

**Problem 1**

**(a)** First, we put in the file `ps4-p1-efa.txt` the Forlan syntax for $M$:

```
{states} A, B, C, D {start state} A {accepting states} A
{transitions}
A, % -> B; A, 0 -> B; B, % -> C; B, 2 -> C; C, % -> D; C, 4 -> D;
D, % -> C; D, 5 -> C; C, % -> B; C, 3 -> B; B, % -> A; B, 1 -> A
```

Then, we proceed as follows:

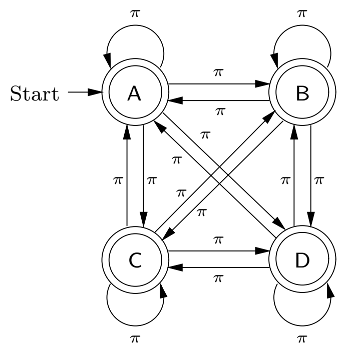
```
- val efa = EFA.input "ps4-p1-efa.txt";
val efa = - : efa
- val lp1 = EFA.findAcceptingLP efa (Str.fromString "012345012345");
val lp1 = - : lp
- LP.output("", lp1);
A, 0 => B, 1 => A, % => B, 2 => C, 3 => B, % => C, 4 => D, 5 => C, % => B, % =>
A, 0 => B, 1 => A, % => B, 2 => C, 3 => B, % => C, 4 => D, 5 => C, % => B, % =>
A
val it = () : unit
- val lp2 = EFA.findAcceptingLP efa (Str.fromString "04320123552");
val lp2 = - : lp
- LP.output("", lp2);
A, 0 => B, % => C, 4 => D, % => C, 3 => B, 2 => C, % => B, % => A, 0 => B, 1 =>
A, % => B, 2 => C, 3 => B, % => C, % => D, 5 => C, % => D, 5 => C, % => B, 2 =>
C, % => B, % => A
val it = () : unit
```

**(b)** First of all, it is easy to see that, for all $q \in Q_M$, $\textbf{emptyClose}_M\,\{q\} = \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\} = \textbf{emptyCloseBackwards}_M\,\{q\}$.

We have that $Q_N = Q_M = \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}$ and $s_N = s_M = \mathsf{A}$.

Next, $A_N = \textbf{emptyCloseBackwards}\,A_M = \textbf{emptyCloseBackwards}\,\{\mathsf{A}\} = \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}$.

Finally, we calculate $T_N$ by working through the non-% transitions of $A_M$. Because of the transition $(\mathsf{A},\mathsf{0},\mathsf{B})$ of $M$, we add to $T_N$ all the elements of $\{\,(q',\mathsf{0},r') \mid q' \in \textbf{emptyCloseBackwards}\,\{\mathsf{A}\}$ and $r' \in \textbf{emptyClose}\,\{\mathsf{B}\}\,\}$, which is equal to the 16-element set $\{\,(q',\mathsf{0},r') \mid q' \in \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}$ and $r' \in \{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}\,\}$. In other words, there is a 0-transition between each pair of states of $N$ (including from a state back to itself). Because $M$ also has a transition labeled 1, $N$ also has a 1-transition between each pair of states. And the same is true for each of 2, 3, 4 and 5—the other elements of $M$'s alphabet. Thus $N$ is




$\pi = 0, 1, 2, 3, 4, 5$

(Between any two states of $N$ (including from a state back to itself), there are transitions labeled 0, 1, 2, 3, 4 and 5. There are a total of $6 \times 16 = 96$ transitions.)

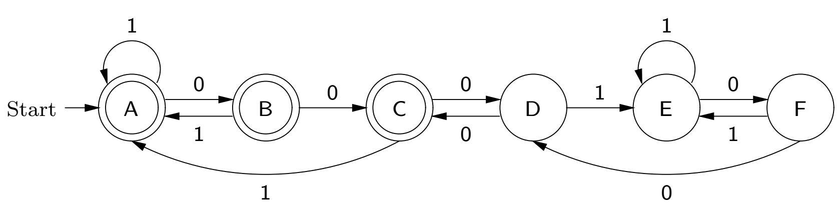
To check that we found the correct NFA, we continue our Forlan session as follows:

```
- val nfa = efaToNFA efa;
val nfa = - : nfa
- NFA.output("", nfa);
{states} A, B, C, D {start state} A {accepting states} A, B, C, D
{transitions}
A, 0 -> A | B | C | D; A, 1 -> A | B | C | D; A, 2 -> A | B | C | D;
A, 3 -> A | B | C | D; A, 4 -> A | B | C | D; A, 5 -> A | B | C | D;
B, 0 -> A | B | C | D; B, 1 -> A | B | C | D; B, 2 -> A | B | C | D;
B, 3 -> A | B | C | D; B, 4 -> A | B | C | D; B, 5 -> A | B | C | D;
C, 0 -> A | B | C | D; C, 1 -> A | B | C | D; C, 2 -> A | B | C | D;
C, 3 -> A | B | C | D; C, 4 -> A | B | C | D; C, 5 -> A | B | C | D;
D, 0 -> A | B | C | D; D, 1 -> A | B | C | D; D, 2 -> A | B | C | D;
D, 3 -> A | B | C | D; D, 4 -> A | B | C | D; D, 5 -> A | B | C | D
val it = () : unit
- NFA.numTransitions nfa;
val it = 96 : int
```

**(c)** $L(M)$ is $\{0, 1, 2, 3, 4, 5\}^*$.

**Problem 2**

**(a)** $M$ is



**(b)** First, we put in the file `ps4-p2-dfa.txt` the Forlan description of $M$:

```
{states} A, B, C, D, E, F {start state} A {accepting states} A, B, C
{transitions}
A, 0 -> B; A, 1 -> A; B, 0 -> C; B, 1 -> A;
C, 0 -> D; C, 1 -> A; D, 0 -> C; D, 1 -> E;
E, 0 -> F; E, 1 -> E; F, 0 -> D; F, 1 -> E
```

Next, we put our testing code in the file `ps4-p2-testing.sml`:

```
(* the function f *)

fun f xs =
      let fun h(us, []) = Set.empty
            | h(us, vs) =
                if Str.prefix(Str.fromString "000", vs)
                then StrSet.union(Set.sing(rev us), h(hd vs :: us, tl vs))
                else h(hd vs :: us, tl vs)
      in h([], xs) end;

(* the function g *)

fun g xs = Set.size(f xs);

(* test for being an element of the language X *)

fun in_X xs = g xs mod 2 = 0;

(* if n >= 0, upto n returns the set of all strings of 0s and 1s of
   length <= n *)

fun upto 0 = StrSet.fromString "%"
  | upto n =
      StrSet.union
      (StrSet.power(StrSet.fromString "0, 1", n),
       upto(n - 1));

(* if n >= 0, boundedTest n dfa assesses dfa using all test data of
   length <= n *)

fun boundedTest n dfa =
      let val alls   = upto n
          val goods  = Set.filter in_X alls
          val bads   = StrSet.minus(alls, goods)
          val acc    = DFA.accepted dfa
          val notAcc = not o acc
      in Set.all acc goods andalso Set.all notAcc bads end;

(* test dfa assesses dfa using all test data of length <= 20 *)
```

```
val test = boundedTest 20;
```

We then proceed as follows:

```
- use "ps4-p2-testing.sml";
[opening ps4-p2-testing.sml]
val f = fn : sym list -> str set
val g = fn : sym list -> int
val in_X = fn : sym list -> bool
val upto = fn : int -> str set
val boundedTest = fn : int -> dfa -> bool
val test = fn : dfa -> bool
val it = () : unit
- val dfa = DFA.input "ps4-p2-dfa.txt";
val dfa = - : dfa
- test dfa;
val it = true : bool
```

**(c)**

**Lemma PS4.2.1**
*(1)* $g\,\% = 0$.

*(2) For all* $x \in \{0,1\}^*$, $g(x1) = g\,x$;

*(3) For all* $x \in \{0,1\}^*$, *if* 00 *is a suffix of* $x$, *then* $g(x0) = g\,x + 1$.

*(4) For all* $x \in \{0,1\}^*$, *if* 00 *is not suffix of* $x$, *then* $g(x0) = g\,x$.

**Proof.**

(1) To show that $f\,\% \subseteq \emptyset$, suppose $y \in f\,\%$. Thus $y \in \{0,1\}^*$ and $y000$ is a prefix of %—contradiction. Thus $y \in \emptyset$. Since $\emptyset \subseteq f\,\%$, it follows that $f\,\% = \emptyset$. Thus $g\,\% = |f\,\%| = |\emptyset| = 0$.

(2) Suppose $x \in \{0,1\}^*$.

To show that $f(x1) \subseteq f\,x$, suppose $y \in f(x1)$. Thus $y \in \{0,1\}^*$ and $y000$ is a prefix of $x1$. Since $y000 \neq x1$, it follows that $y000$ is a prefix of $x$, and thus that $y \in f\,x$.

To show that $f\,x \subseteq f(x1)$, suppose $y \in f\,x$. Thus $y \in \{0,1\}^*$ and $y000$ is a prefix of $x$. Hence $y000$ is a prefix of $x1$, so that $y \in f(x1)$.

Thus $f(x1) = f\,x$. Finally, $g(x1) = |f(x1)| = |f\,x| = g\,x$.

(3) Suppose $x \in \{0,1\}^*$ and 00 is a suffix of $x$. Thus $x = z00$ for some $z \in \{0,1\}^*$.

To show that $f(x0) \subseteq f\,x \cup \{z\}$, suppose $y \in f(x0)$. Thus $y \in \{0,1\}^*$ and $y000$ is a prefix of $x0 = z000$. There are two cases to consider. (1) Suppose $y000$ is prefix of $z00 = x$. Then $y \in f\,x \subseteq f\,x \cup \{z\}$. (2) Suppose $y000 = z000$. Then $y = z \in f\,x \cup \{z\}$.

To show that $f\,x \cup \{z\} \subseteq f(x0)$, suppose $y \in f\,x \cup \{z\}$. There are two cases to consider. (1) Suppose $y \in f\,x$. Then $y \in \{0,1\}^*$ and $y000$ is a prefix of $x$, so that $y000$ is also a prefix of $x0$.

Thus $y \in f(x0)$. (2) Suppose $y = z$. Then $y000 = z000 = x0$, so that $y000$ is a prefix of $x0$, and thus $y \in f(x0)$.

Thus $f(x0) = f\,x \cup \{z\}$. Because $x = z00$, we have that $z \notin f\,x$—since otherwise we would have $z000$ is a prefix of $x = z00$, which is impossible.

Finally, $g(x0) = |f(x0)| = |f\,x \cup \{z\}| = |f\,x| + 1 = g\,x + 1$, because $z \notin f\,x$.

(4) Suppose $x \in \{0,1\}^*$ and $00$ is not a suffix of $x$.

To show that $f(x0) \subseteq f\,x$, suppose $y \in f(x0)$. Thus $y \in \{0,1\}^*$ and $y000$ is a prefix of $x0$. Because $00$ is not a suffix of $x$, it follows that $000$ is not a suffix of $x0$. Thus $y000 \neq x0$, so that $y000$ is a prefix of $x$. Hence $y \in f\,x$.

To show that $f\,x \subseteq f(x0)$, suppose $y \in f\,x$. Thus $y \in \{0,1\}^*$ and $y000$ is a prefix of $x$. Hence $y000$ is a prefix of $x0$, so that $y \in f\,(x0)$.

Thus $f(x0) = f\,x$. Finally, $g(x0) = |f(x0)| = |f\,x| = g\,x$.

□

**Lemma PS4.2.2**

*(1) $\% \in X$.*

*(2) For all $w \in \{0,1\}^*$, if $w \in X$, then $w1 \in X$.*

*(3) For all $w \in \{0,1\}^*$, if $w \in X$ and $00$ is a suffix of $w$, then $w0 \notin X$.*

*(4) For all $w \in \{0,1\}^*$, if $w \in X$ and $00$ is not a suffix of $w$, then $w0 \in X$.*

*(5) For all $w \in \{0,1\}^*$, if $w \notin X$, then $w1 \notin X$.*

*(6) For all $w \in \{0,1\}^*$, if $w \notin X$ and $00$ is a suffix of $w$, then $w0 \in X$.*

*(7) For all $w \in \{0,1\}^*$, if $w \notin X$ and $00$ is not a suffix of $w$, then $w0 \notin X$.*

**Proof.**

(1) By Lemma PS4.2.1(1), we have that $g\,\% = 0$ is even. Thus $\% \in X$.

(2) Suppose $w \in \{0,1\}^*$ and $w \in X$. Then $g\,w$ is even, so that $g(w1) = g\,w$ is even, by Lemma PS4.2.1(2). Thus $w1 \in X$.

(3) Suppose $w \in \{0,1\}^*$, $w \in X$ and $00$ is a suffix of $w$. Thus $g\,w$ is even, so that $g(w0) = g\,w + 1$ is odd, by Lemma PS4.2.1(3). Thus $w0 \notin X$.

(4) Suppose $w \in \{0,1\}^*$, $w \in X$ and $00$ is not a suffix of $w$. Thus $g\,w$ is even, so that $g(w0) = g\,w$ is even, by Lemma PS4.2.1(4). Thus $w0 \in X$.

(5) Suppose $w \in \{0,1\}^*$ and $w \notin X$. Then $g\,w$ is odd, so that $g(w1) = g\,w$ is odd, by Lemma PS4.2.1(2). Thus $w1 \notin X$.

(6) Suppose $w \in \{0,1\}^*$, $w \notin X$ and $00$ is a suffix of $w$. Thus $g\,w$ is odd, so that $g(w0) = g\,w + 1$ is even, by Lemma PS4.2.1(3). Thus $w0 \in X$.

(7) Suppose $w \in \{0,1\}^*$, $w \notin X$ and $00$ is not a suffix of $w$. Thus $g\,w$ is odd, so that $g(w0) = g\,w$ is odd, by Lemma PS4.2.1(4). Thus $w0 \notin X$.

□

**Lemma PS4.2.3**

*(A) For all $w \in \Lambda_{\mathsf{A}}$, $w \in X$ and $0$ is not a suffix of $w$.*

*(B) For all $w \in \Lambda_{\mathsf{B}}$, $w \in X$ and $0$, but not $00$, is a suffix of $w$.*

*(C) For all $w \in \Lambda_{\mathsf{C}}$, $w \in X$ and $00$ is a suffix of $w$.*

*(D) For all $w \in \Lambda_{\mathsf{D}}$, $w \notin X$ and $00$ is a suffix of $w$.*

*(E) For all $w \in \Lambda_{\mathsf{E}}$, $w \notin X$ and $1$ is a suffix of $w$.*

*(F) For all $w \in \Lambda_{\mathsf{F}}$, $w \notin X$ and $0$, but not $00$, is a suffix of $w$.*

**Proof.** We proceed by induction on $\Lambda$. There are 13 (1 plus the number of transitions) parts to show.

**(empty string)** We must show that $\% \in X$ and $0$ is not a suffix of $\%$. The latter property is obvious, and the former follows by Lemma PS4.2.2(1).

$(\mathsf{A}, 0 \to \mathsf{B})$ Suppose $w \in \Lambda_{\mathsf{A}}$, and assume the inductive hypothesis: $w \in X$ and $0$ is not a suffix of $w$. We must show that $w0 \in X$ and $0$, but not $00$, is a suffix of $w0$. Clearly $0$ is a suffix of $w0$. And since $0$ is not a suffix of $w$, we have that $00$ is not a suffix of $w0$. Since $w \in X$ and $00$ is not a suffix of $w$, Lemma PS4.2.2(4) tells us that $w0 \in X$.

$(\mathsf{A}, 1 \to \mathsf{A})$ Suppose $w \in \Lambda_{\mathsf{A}}$, and assume the inductive hypothesis: $w \in X$ and $0$ is not a suffix of $w$. We must show that $w1 \in X$ and $0$ is not a suffix of $w1$. The latter property is obvious. And the former property holds by Lemma PS4.2.2(2).

$(\mathsf{B}, 0 \to \mathsf{C})$ Suppose $w \in \Lambda_{\mathsf{B}}$, and assume the inductive hypothesis: $w \in X$ and $0$, but not $00$, is a suffix of $w$. We must show that $w0 \in X$ and $00$ is a suffix of $w0$. Since $0$ is a suffix of $w$, we have that $00$ is a suffix of $w0$. Because $00$ is not a suffix of $w$, Lemma PS4.2.2(4) tells us that $w0 \in X$.

$(\mathsf{B}, 1 \to \mathsf{A})$ Suppose $w \in \Lambda_{\mathsf{B}}$, and assume the inductive hypothesis: $w \in X$ and $0$, but not $00$, is a suffix of $w$. We must show that $w1 \in X$ and $0$ is not a suffix of $w1$. The latter property is obvious, and the former follows by Lemma PS4.2.2(2).

$(\mathsf{C}, 0 \to \mathsf{D})$ Suppose $w \in \Lambda_{\mathsf{C}}$, and assume the inductive hypothesis: $w \in X$ and $00$ is a suffix of $w$. We must show that $w0 \notin X$ and $00$ is a suffix of $w0$. The latter property holds, since $0$ is a suffix of $w$. Because $00$ is a suffix of $w$, Lemma PS4.2.2(3) tells us that $w0 \notin X$.

$(\mathsf{C}, 1 \to \mathsf{A})$ Suppose $w \in \Lambda_{\mathsf{C}}$, and assume the inductive hypothesis: $w \in X$ and $00$ is a suffix of $w$. We must show that $w1 \in X$ and $0$ is not a suffix of $w1$. The latter property is obvious. And the former follows by Lemma PS4.2.2(2).

- $(\mathsf{D}, \mathsf{0} \to \mathsf{C})$ Suppose $w \in \Lambda_{\mathsf{D}}$, and assume the inductive hypothesis: $w \notin X$ and $\mathsf{00}$ is a suffix of $w$. We must show that $w\mathsf{0} \in X$ and $\mathsf{00}$ is a suffix of $w\mathsf{0}$. The latter property holds, since $\mathsf{0}$ is a suffix of $w$. Because $\mathsf{00}$ is a suffix of $w$, Lemma PS4.2.2(6) tells us that $w\mathsf{0} \in X$.

- $(\mathsf{D}, \mathsf{1} \to \mathsf{E})$ Suppose $w \in \Lambda_{\mathsf{D}}$, and assume the inductive hypothesis: $w \notin X$ and $\mathsf{00}$ is a suffix of $w$. We must show that $w\mathsf{1} \notin X$ and $\mathsf{1}$ is a suffix of $w\mathsf{1}$. The latter property obviously holds. And the former follows by Lemma PS4.2.2(5).

- $(\mathsf{E}, \mathsf{0} \to \mathsf{F})$ Suppose $w \in \Lambda_{\mathsf{E}}$, and assume the inductive hypothesis: $w \notin X$ and $\mathsf{1}$ is a suffix of $w$. We must show that $w\mathsf{0} \notin X$ and $\mathsf{0}$, but not $\mathsf{00}$, is a suffix of $w\mathsf{0}$. Clearly $\mathsf{0}$ is a suffix of $w\mathsf{0}$. Because $\mathsf{0}$ is not a suffix of $w$, it follows that $\mathsf{00}$ is not a suffix of $w\mathsf{0}$. Because $\mathsf{00}$ is not a suffix of $w$, Lemma PS4.2.2(7) tells us that $w\mathsf{0} \notin X$.

- $(\mathsf{E}, \mathsf{1} \to \mathsf{E})$ Suppose $w \in \Lambda_{\mathsf{E}}$, and assume the inductive hypothesis: $w \notin X$ and $\mathsf{1}$ is a suffix of $w$. We must show that $w\mathsf{1} \notin X$ and $\mathsf{1}$ is a suffix of $w\mathsf{1}$. The latter property obviously holds. And the former follows by Lemma PS4.2.2(5).

- $(\mathsf{F}, \mathsf{0} \to \mathsf{D})$ Suppose $w \in \Lambda_{\mathsf{F}}$, and assume the inductive hypothesis: $w \notin X$ and $\mathsf{0}$, but not $\mathsf{00}$, is a suffix of $w$. We must show that $w\mathsf{0} \notin X$ and $\mathsf{00}$ is a suffix of $w\mathsf{0}$. Since $\mathsf{0}$ is a suffix of $w$, we have that $\mathsf{00}$ is a suffix of $w\mathsf{0}$. Because $\mathsf{00}$ is not a suffix of $w$, Lemma PS4.2.2(7) tells us that $w\mathsf{0} \notin X$.

- $(\mathsf{F}, \mathsf{1} \to \mathsf{E})$ Suppose $w \in \Lambda_{\mathsf{F}}$, and assume the inductive hypothesis: $w \notin X$ and $\mathsf{0}$, but not $\mathsf{00}$, is a suffix of $w$. We must show that $w\mathsf{1} \notin X$ and $\mathsf{1}$ is a suffix of $w\mathsf{1}$. The latter property is obvious. And the former follows by Lemma PS4.2.2(5).

□

**Proposition PS4.2.4**
$L(M) = X$.

**Proof.** We show that $L(M) \subseteq X \subseteq L(M)$.

- $(L(M) \subseteq X)$ Suppose $w \in L(M)$. Because $A_M = \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$, we have that $w \in L(M) = \Lambda_{\mathsf{A}} \cup \Lambda_{\mathsf{B}} \cup \Lambda_{\mathsf{C}}$. Thus, by Lemma PS4.2.3(A)–(C), we have that $w \in X$.

- $(X \subseteq L(M))$ Suppose $w \in X$. Since $X \subseteq \{\mathsf{0}, \mathsf{1}\}^*$, we have that $w \in \{\mathsf{0}, \mathsf{1}\}^*$. Suppose, toward a contradiction, that $w \notin L(M)$. Because $w \notin L(M) = \Lambda_{\mathsf{A}} \cup \Lambda_{\mathsf{B}} \cup \Lambda_{\mathsf{C}}$ and $w \in \{\mathsf{0}, \mathsf{1}\}^* = (\textbf{alphabet}\, M)^* = \Lambda_{\mathsf{A}} \cup \Lambda_{\mathsf{B}} \cup \Lambda_{\mathsf{C}} \cup \Lambda_{\mathsf{D}} \cup \Lambda_{\mathsf{E}} \cup \Lambda_{\mathsf{F}}$, we must have that $w \in \Lambda_{\mathsf{D}} \cup \Lambda_{\mathsf{E}} \cup \Lambda_{\mathsf{F}}$. But then Lemma PS4.2.3(D)–(F) tells us that $w \notin X$—contradiction. Thus $w \in L(M)$.

□