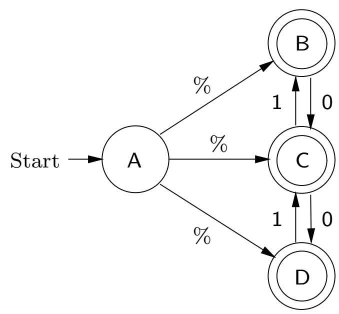
CS 516—Software Foundations via Formal Languages—Spring 2022

Problem Set 5

Model Answers

Problem 1

Let M be the finite automaton



of Problem 2 of Problem Set 4. The model answers to Problem 3 of Problem Set 4 proved that $L(M) = X$, and we put the expression of M in Forlan's concrete syntax in the file `ps5-p1-fa` (see the course website):

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

We load M into Forlan, calling it `fa`:

```
- val fa = FA.input "ps5-p1-fa";
val fa = - : fa
```

We use the function `faToRegPerms` to convert `fa` to a regular expression, using weak simplification as the simplification function. This tries eliminating states in all possible orders, and uses the permutation on states that results in the simplest possible regular expression. We bind the result to `reg'`, and display it:

```
- val reg' = faToRegPerms (NONE, Reg.weaklySimplify) fa;
val reg' = - : reg
- Reg.output("", reg');
% + (% + 0 + 1)(01 + 10)*(% + 0 + 1)
val it = () : unit
```

Finally, we locally simplify `reg'`, yielding the regular expression `reg`, and display `reg`:

```
- val reg = #2 (Reg.locallySimplify (NONE, Reg.obviousSubset) reg');
```

```
val reg = - : reg
- Reg.output("", reg);
(% + 0 + 1)(01 + 10)*(% + 0 + 1)
val it = () : unit
```

Problem 2

First, we put the definition

```
val faToDFA = nfaToDFA o efaToNFA o faToEFA;

fun subst(fa, x, y) =
      if FA.accepted fa x
      then FA.union
           (injDFAToFA (DFA.minus(faToDFA fa, faToDFA(strToFA x))),
            strToFA y)
      else fa;
```

of `subst` in the file `ps5-p2.sml` (see the course website), and load it into Forlan:

```
- use "ps5-p2.sml";
[opening ps5-p2.sml]
val faToDFA = fn : fa -> dfa
val subst = fn : fa * str * str -> fa
val it = () : unit
```

Next, we put the definition

```
fun test(reg, x, y, reg') =
      let val reg  = Reg.fromString reg
          val x    = Str.fromString x
          val y    = Str.fromString y
          val reg' = Reg.fromString reg'

          val fa  = subst(regToFA reg, x, y)
          val fa' = regToFA reg'
      in DFA.equivalent(faToDFA fa, faToDFA fa') end;
```

of our testing function `test` in the file `ps5-p2-testing.sml`, (see the course website) and load it into Forlan:

```
- use "ps5-p2-testing.sml";
[opening ps5-p2-testing.sml]
val test = fn : string * string * string * string -> bool
val it = () : unit
```

Finally, we run tests corresponding to the examples from the description of Problem 2:

```
- test("01 + 10", "01", "11", "11 + 10");
val it = true : bool
- test("01 + 10", "01", "01", "01 + 10");
```

```
val it = true : bool
- test("01 + 10", "01", "10", "10");
val it = true : bool
- test("01 + 10", "11", "12", "01 + 10");
val it = true : bool
```

Problem 3

First, we put the definition

```
fun superAccepted fa w =
      let val start     = FA.startState fa
          val accepting = FA.acceptingStates fa
          val qs        = FA.processStr fa (Set.sing start, w)
      in Set.isNonEmpty qs andalso SymSet.subset(qs, accepting) end;
```

of `superAccepted` in the file `ps5-p3.sml` (see the course website), and load it into Forlan:

```
- use "ps5-p3.sml";
[opening ps5-p3.sml]
val superAccepted = fn : fa -> str -> bool
val it = () : unit
```

Next, we put the definition

```
fun test reg w =
      let val fa = regToFA reg
      in FA.accepted fa w = superAccepted fa w end;
```

of `test` in the file `ps5-p3-testing.sml` (see the course website), and load it into Forlan:

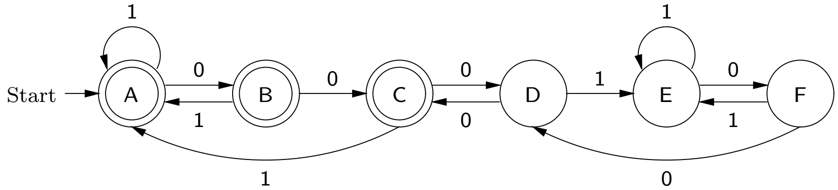
```
- use "ps5-p3-testing.sml";
[opening ps5-p3-testing.sml]
val test = fn : reg -> str -> bool
val it = () : unit
```

Finally, we show that arguments 0^* and 0 suffice to make `test` return `false`:

```
- test (Reg.fromString "0*") (Str.fromString "0");
val it = false : bool
```

Problem 4

(a) M is



(b)

Lemma PS5.4.1
(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 $x0$. Then $y000 \neq x0$, since otherwise 00 would be a suffix of x. Thus $y000$ 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 PS5.4.2
(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 PS5.4.1(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 PS5.4.1(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 PS5.4.1(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 PS5.4.1(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 PS5.4.1(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 PS5.4.1(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 PS5.4.1(4). Thus $w0 \notin X$.

□

Lemma PS5.4.3

(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 Λ. There are 13 (1 plus the number of transitions) parts to show. Note that whenever we assume $w \in \Lambda_q$, for some $q \in Q_M$, we have that $w \in (\textbf{alphabet}\, M)^* = \{0,1\}^*$.

(empty string) We must show that $\% \in X$ and $\mathsf{0}$ is not a suffix of $\%$. The latter property is obvious, and the former follows by Lemma PS5.4.2(1).

$(\mathsf{A}, \mathsf{0} \to \mathsf{B})$ Suppose $w \in \Lambda_{\mathsf{A}}$, and assume the inductive hypothesis: $w \in X$ and $\mathsf{0}$ is not a suffix of w. We must show that $w\mathsf{0} \in X$ and $\mathsf{0}$, but not $\mathsf{00}$, is a suffix of $w\mathsf{0}$. Clearly $\mathsf{0}$ is a suffix of $w\mathsf{0}$. And since $\mathsf{0}$ is not a suffix of w, we have that $\mathsf{00}$ is not a suffix of $w\mathsf{0}$. Since $w \in X$ and $\mathsf{00}$ is not a suffix of w, Lemma PS5.4.2(4) tells us that $w\mathsf{0} \in X$.

$(\mathsf{A}, \mathsf{1} \to \mathsf{A})$ Suppose $w \in \Lambda_{\mathsf{A}}$, and assume the inductive hypothesis: $w \in X$ and $\mathsf{0}$ is not a suffix of w. We must show that $w\mathsf{1} \in X$ and $\mathsf{0}$ is not a suffix of $w\mathsf{1}$. The latter property is obvious. And the former property holds by Lemma PS5.4.2(2).

$(\mathsf{B}, \mathsf{0} \to \mathsf{C})$ Suppose $w \in \Lambda_{\mathsf{B}}$, and assume the inductive hypothesis: $w \in X$ and $\mathsf{0}$, but not $\mathsf{00}$, is a suffix of w. We must show that $w\mathsf{0} \in X$ and $\mathsf{00}$ is a suffix of $w\mathsf{0}$. 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 PS5.4.2(4) tells us that $w\mathsf{0} \in X$.

$(\mathsf{B}, \mathsf{1} \to \mathsf{A})$ Suppose $w \in \Lambda_{\mathsf{B}}$, and assume the inductive hypothesis: $w \in X$ and $\mathsf{0}$, but not $\mathsf{00}$, is a suffix of w. We must show that $w\mathsf{1} \in X$ and $\mathsf{0}$ is not a suffix of $w\mathsf{1}$. The latter property is obvious, and the former follows by Lemma PS5.4.2(2).

$(\mathsf{C}, \mathsf{0} \to \mathsf{D})$ Suppose $w \in \Lambda_{\mathsf{C}}$, and assume the inductive hypothesis: $w \in X$ and $\mathsf{00}$ is a suffix of w. We must show that $w\mathsf{0} \notin X$ and $\mathsf{00}$ is a suffix of $w\mathsf{0}$. The latter property holds, since $\mathsf{0}$ is a suffix of w. Because $\mathsf{00}$ is a suffix of w, Lemma PS5.4.2(3) tells us that $w\mathsf{0} \notin X$.

$(\mathsf{C}, \mathsf{1} \to \mathsf{A})$ Suppose $w \in \Lambda_{\mathsf{C}}$, and assume the inductive hypothesis: $w \in X$ and $\mathsf{00}$ is a suffix of w. We must show that $w\mathsf{1} \in X$ and $\mathsf{0}$ is not a suffix of $w\mathsf{1}$. The latter property is obvious. And the former follows by Lemma PS5.4.2(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 PS5.4.2(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 PS5.4.2(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 PS5.4.2(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 PS5.4.2(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 PS5.4.2(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 PS5.4.2(5).

□

Proposition PS5.4.4
$L(M) = X$.

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 PS5.4.3(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 PS5.4.3(D)–(F) tells us that $w \notin X$—contradiction. Thus $w \in L(M)$.

□