CIS 570 — Introduction to Formal Language Theory — Fall 2007

Exercise Set 5

Model Answers

Exercise 1

First, we let the DFA M' be **determSimplify**$(M, \emptyset)$. M' is the same as M. Next, we construct the set X of pairs of states of M', as follows.

First, we add to X all pairs consisting of an accepting state and a non-accepting state: (A, B), (B, A), (A, D), (D, A), (B, C), (C, B), (C, D) and (D, C). Now we must handle each of these 8 pairs.

Since there are no $\mathsf{0}$-transitions leading into B, nothing can be added to X using (A, B), (B, A) and $\mathsf{0}$-transitions. Since there are no $\mathsf{1}$-transitions leading into A, nothing can be added to X using (A, B), (B, A) and $\mathsf{1}$-transitions.

Since (A, D) and (D, A) are in X, and $(\mathsf{A}, \mathsf{0}, \mathsf{A})$, $(\mathsf{C}, \mathsf{0}, \mathsf{A})$, $(\mathsf{B}, \mathsf{0}, \mathsf{D})$ and $(\mathsf{D}, \mathsf{0}, \mathsf{D})$ are the $\mathsf{0}$-transitions leading into A and D, we would add the pairs (A, B), (B, A), (A, D), (D, A), (B, C), (C, B), (C, D) and (D, C) to X, if they weren't already there. Since there are no $\mathsf{1}$-transitions leading into A, nothing can be added to X using (A, D), (D, A) and $\mathsf{1}$-transitions.

Since there are no $\mathsf{0}$-transitions leading into B, nothing can be added to X using (B, C), (C, B) and $\mathsf{0}$-transitions. Since (B, C) and (C, B) are in X, and $(\mathsf{A}, \mathsf{1}, \mathsf{B})$, $(\mathsf{B}, \mathsf{1}, \mathsf{C})$ and $(\mathsf{D}, \mathsf{1}, \mathsf{C})$ are the $\mathsf{1}$-transitions leading into B and C, we would add the pairs (A, B), (B, A), (A, D) and (D, A) to X, if they weren't already there.

Since there are no $\mathsf{0}$-transitions leading into C, nothing can be added to X using (C, D), (D, C) and $\mathsf{0}$-transitions. Since (C, D) and (D, C) are in X, and $(\mathsf{B}, \mathsf{1}, \mathsf{C})$, $(\mathsf{D}, \mathsf{1}, \mathsf{C})$ and $(\mathsf{C}, \mathsf{1}, \mathsf{D})$ are the $\mathsf{1}$-transitions leading into C and D, we would add the pairs (B, C), (C, B), (D, C) and (C, D) to X, if they weren't already there.

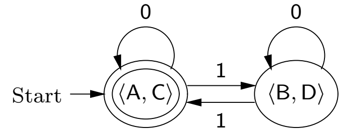
We have now handled all of the elements of X that were initially added to X, using rules (1) and (2). And no new elements were added to X, and so we have that X consists of the following 8 pairs: (A, B), (A, D), (B, A), (B, C), (C, B), (C, D), (D, A) and (D, C). Thus the set Y consists of the following 8 pairs: (A, A), (A, C), (B, B), (B, D), (C, A), (C, C), (D, B) and (D, D). Hence the set Z consists of the following equivalence classes: $\{\mathsf{A}, \mathsf{C}\}$ and $\{\mathsf{B}, \mathsf{D}\}$.

Hence N has the following states: $\langle \mathsf{A}, \mathsf{C} \rangle$ and $\langle \mathsf{B}, \mathsf{D} \rangle$. Since A is the start state of M', we have that the start state of N is $\langle \mathsf{A}, \mathsf{C} \rangle$. Since A and C are the accepting states of M', we have that $\langle \mathsf{A}, \mathsf{C} \rangle$ is the only accepting state of N. It remains to compute the transitions of N.

Since $\{\mathsf{A}, \mathsf{C}\} \in Z$, and $[\delta_{M'}(\mathsf{A}, \mathsf{0})] = [\mathsf{A}] = \{\mathsf{A}, \mathsf{C}\}$, we have that $(\langle \mathsf{A}, \mathsf{C} \rangle, \mathsf{0}, \langle \mathsf{A}, \mathsf{C} \rangle) \in T_N$. Since $\{\mathsf{A}, \mathsf{C}\} \in Z$, and $[\delta_{M'}(\mathsf{A}, \mathsf{1})] = [\mathsf{B}] = \{\mathsf{B}, \mathsf{D}\}$, we have that $(\langle \mathsf{A}, \mathsf{C} \rangle, \mathsf{1}, \langle \mathsf{B}, \mathsf{D} \rangle) \in T_N$.

Since $\{\mathsf{B}, \mathsf{D}\} \in Z$, and $[\delta_{M'}(\mathsf{B}, \mathsf{0})] = [\mathsf{D}] = \{\mathsf{B}, \mathsf{D}\}$, we have that $(\langle \mathsf{B}, \mathsf{D} \rangle, \mathsf{0}, \langle \mathsf{B}, \mathsf{D} \rangle) \in T_N$. Since $\{\mathsf{B}, \mathsf{D}\} \in Z$, and $[\delta_{M'}(\mathsf{B}, \mathsf{1})] = [\mathsf{C}] = \{\mathsf{A}, \mathsf{C}\}$, we have that $(\langle \mathsf{B}, \mathsf{D} \rangle, \mathsf{1}, \langle \mathsf{A}, \mathsf{C} \rangle) \in T_N$.

Here is a drawing of N:



To check that our final answer is correct, we put the text

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

in the file es5-ex1-dfa. Then we invoke Forlan and proceed as follows:

```
- val dfa = DFA.input "es5-ex1-dfa";
val dfa = - : dfa
- val dfa' = DFA.minimize dfa;
val dfa' = - : dfa
- DFA.output("", dfa');
{states}
<A,C>, <B,D>
{start state}
<A,C>
{accepting states}
<A,C>
{transitions}
<A,C>, 0 -> <A,C>; <A,C>, 1 -> <B,D>; <B,D>, 0 -> <B,D>; <B,D>, 1 -> <A,C>
val it = () : unit
```

It is easy to check that the outputted DFA is N.

Exercise 2

Since

$$\begin{aligned}\mathbf{Btw}(1,1,0) &= \{\%, \mathsf{1}\},\\ \mathbf{Btw}(1,2,0) &= \{\mathsf{0}\},\\ \mathbf{Btw}(2,1,0) &= \{\mathsf{1}\},\\ \mathbf{Btw}(2,2,0) &= \{\%, \mathsf{0}\},\end{aligned}$$

we have that

$$\begin{aligned}\mathbf{btw}(1,1,0) &= \mathit{simp}(\% + \mathsf{1}) = \% + \mathsf{1},\\ \mathbf{btw}(1,2,0) &= \mathit{simp}(\mathsf{0}) = \mathsf{0},\\ \mathbf{btw}(2,1,0) &= \mathit{simp}(\mathsf{1}) = \mathsf{1},\\ \mathbf{btw}(2,2,0) &= \mathit{simp}(\% + \mathsf{0}) = \% + \mathsf{0}.\end{aligned}$$

Thus

$$
\begin{aligned}
\mathbf{btw}(1,2,1) &= \mathit{simp}(\mathbf{btw}(1,2,0) + \mathbf{btw}(1,1,0)\,\mathbf{btw}(1,1,0)^*\,\mathbf{btw}(1,2,0)) \\
&= \mathit{simp}(\mathsf{0} + (\% + \mathsf{1})(\% + \mathsf{1})^*\mathsf{0}) \\
&= \mathsf{1}^*\mathsf{0}, \\
\mathbf{btw}(2,2,1) &= \mathit{simp}(\mathbf{btw}(2,2,0) + \mathbf{btw}(2,1,0)\,\mathbf{btw}(1,1,0)^*\,\mathbf{btw}(1,2,0)) \\
&= \mathit{simp}((\% + \mathsf{0}) + \mathsf{1}(\% + \mathsf{1})^*\mathsf{0}) \\
&= \% + \mathsf{0} + \mathsf{11}^*\mathsf{0}, \\
\mathbf{btw}(1,2,2) &= \mathit{simp}(\mathbf{btw}(1,2,1) + \mathbf{btw}(1,2,1)\,\mathbf{btw}(2,2,1)^*\,\mathbf{btw}(2,2,1)) \\
&= \mathit{simp}(\mathsf{1}^*\mathsf{0} + (\mathsf{1}^*\mathsf{0})(\% + \mathsf{0} + \mathsf{11}^*\mathsf{0})^*(\% + \mathsf{0} + \mathsf{11}^*\mathsf{0})) \\
&= \mathsf{1}^*\mathsf{0}(\mathsf{0} + \mathsf{11}^*\mathsf{0})^*, \\
\alpha &= \mathit{simp}(\mathbf{btw}(1,2,2)) \\
&= \mathit{simp}(\mathsf{1}^*\mathsf{0}(\mathsf{0} + \mathsf{11}^*\mathsf{0})^*) \\
&= \mathsf{1}^*\mathsf{0}(\mathsf{0} + \mathsf{11}^*\mathsf{0})^*.
\end{aligned}
$$

Here is the Forlan transcript showing the above simplifications:

```
- val simp = Reg.simplify Reg.weakSubset;
val simp = fn : reg -> reg
- fun outSimp s = Reg.output("", simp(Reg.fromString s));
val outSimp = fn : string -> unit
- outSimp "% + 1";
% + 1
val it = () : unit
- outSimp "0";
0
val it = () : unit
- outSimp "1";
1
val it = () : unit
- outSimp "% + 0";
% + 0
val it = () : unit
- outSimp "0 + (%+1)(%+1)*0";
1*0
val it = () : unit
- outSimp "(%+0) + 1(%+1)*0";
% + 0 + 11*0
val it = () : unit
- outSimp "1*0 + (1*0)(%+0+11*0)*(%+0+11*0)";
1*0(0 + 11*0)*
val it = () : unit
- outSimp "1*0(0 + 11*0)*";
1*0(0 + 11*0)*
val it = () : unit
```

To check that our final answer is correct, we put the text

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

in the file `es5-ex2-fa`, and then proceed as follows.

```
- val fa = FA.input "es5-ex2-fa";
val fa = - : fa
- val reg = faToReg simp fa;
val reg = - : reg
- Reg.output("", reg);
1*0(0 + 11*0)*
val it = () : unit
```

Exercise 3

(a) Define functions **HasSuf** $\in \{0,1,2\}^* \to$ **Lan**, **HasNotSuf** $\in \{0,1,2\}^* \to$ **Lan**, **HasPref** $\in \{0,1,2\}^* \to$ **Lan**, **HasNotPref** $\in \{0,1,2\}^* \to$ **Lan** and **NotSur** $\in \{0,1,2\}^* \times \{0,1,2\}^* \times \{0,1,2\}^* \to$ **Lan** by:

- for all $x \in \{0,1,2\}^*$, $\mathbf{HasSuf}(x) = \{\, w \in \{0,1,2\}^* \mid x \text{ is a suffix of } w \,\}$;
- for all $x \in \{0,1,2\}^*$, $\mathbf{HasNotSuf}(x) = \{\, w \in \{0,1,2\}^* \mid x \text{ is not a suffix of } w \,\}$;
- for all $x \in \{0,1,2\}^*$, $\mathbf{HasPref}(x) = \{\, w \in \{0,1,2\}^* \mid x \text{ is a prefix of } w \,\}$;
- for all $x \in \{0,1,2\}^*$, $\mathbf{HasNotPref}(x) = \{\, w \in \{0,1,2\}^* \mid x \text{ is not a prefix of } w \,\}$;
- for all $x, y, z \in \{0,1,2\}^*$, $\mathbf{NotSur}(x, y, z) = \{\, w \in \{0,1,2\}^* \mid$ there are $u, v \in \{0,1,2\}^*$ such that $w = uyv$ and either x is not a suffix of u or z is not a prefix of $v \,\}$.

Lemma ES5.3.1

(1) For all $x \in \{0,1,2\}^*$, $\mathbf{HasSuf}(x) = \{0,1,2\}^*\,\{x\}$.

(2) For all $x \in \{0,1,2\}^*$, $\mathbf{HasNotSuf}(x) = \{0,1,2\}^* - \mathbf{HasSuf}(x)$.

(3) For all $x \in \{0,1,2\}^*$, $\mathbf{HasPref}(x) = \{x\}\,\{0,1,2\}^*$.

(4) For all $x \in \{0,1,2\}^*$, $\mathbf{HasNotPref}(x) = \{0,1,2\}^* - \mathbf{HasPref}(x)$.

(5) For all $x, y, z \in \{0,1,2\}^*$, $\mathbf{NotSur}(x, y, z) = \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^* \cup \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z)$.

(6) *For all* $x, y, z \in \{0,1,2\}^*$, $\mathbf{Sur}(x,y,z) = \{0,1,2\}^* - \mathbf{NotSur}(x,y,z)$.

Proof.

(1) Suppose $x \in \{0,1,2\}^*$. We must show that $\mathbf{HasSuf}(x) = \{0,1,2\}^*\,\{x\}$. It will suffice to show that $\mathbf{HasSuf}(x) \subseteq \{0,1,2\}^*\,\{x\} \subseteq \mathbf{HasSuf}(x)$.

Suppose $w \in \mathbf{HasSuf}(x)$. Then $w \in \{0,1,2\}^*$ and x is a suffix of w. Thus $w = ux$ for some $u \in \{0,1,2\}^*$, so that $w = ux \in \{0,1,2\}^*\,\{x\}$.

Suppose $w \in \{0,1,2\}^*\,\{x\}$. Then $w = ux$ for some $u \in \{0,1,2\}^*$. Hence x is a suffix of w, so that $w \in \mathbf{HasSuf}(x)$.

(2) Suppose $x \in \{0,1,2\}^*$. We must show that $\mathbf{HasNotSuf}(x) = \{0,1,2\}^* - \mathbf{HasSuf}(x)$. It will suffice to show that $\mathbf{HasNotSuf}(x) \subseteq \{0,1,2\}^* - \mathbf{HasSuf}(x) \subseteq \mathbf{HasNotSuf}(x)$.

Suppose $w \in \mathbf{HasNotSuf}(x)$. Then $w \in \{0,1,2\}^*$ and x is not a suffix of w. Suppose, toward a contradiction, that $w \in \mathbf{HasSuf}(x)$. Then x is a suffix of w—contradiction. Thus $w \notin \mathbf{HasSuf}(x)$, completing the proof that $w \in \{0,1,2\}^* - \mathbf{HasSuf}(x)$.

Suppose $w \in \{0,1,2\}^* - \mathbf{HasSuf}(x)$. Then $w \in \{0,1,2\}^*$ and $w \notin \mathbf{HasSuf}(x)$. Suppose, toward a contradiction, that x is a suffix of w. Then $w \in \mathbf{HasSuf}(x)$—contradiction. Hence x is not a suffix of w, completing the proof that $w \in \mathbf{HasNotSuf}(x)$.

(3) Suppose $x \in \{0,1,2\}^*$. We must show that $\mathbf{HasPref}(x) = \{x\}\,\{0,1,2\}^*$. It will suffice to show that $\mathbf{HasPref}(x) \subseteq \{x\}\,\{0,1,2\}^* \subseteq \mathbf{HasPref}(x)$.

Suppose $w \in \mathbf{HasPref}(x)$. Then $w \in \{0,1,2\}^*$ and x is a prefix of w. Thus $w = xu$ for some $u \in \{0,1,2\}^*$, so that $w = xu \in \{x\}\,\{0,1,2\}^*$.

Suppose $w \in \{x\}\,\{0,1,2\}^*$. Then $w = xu$ for some $u \in \{0,1,2\}^*$. Hence x is a prefix of w, so that $w \in \mathbf{HasPref}(x)$.

(4) Suppose $x \in \{0,1,2\}^*$. We must show that $\mathbf{HasNotPref}(x) = \{0,1,2\}^* - \mathbf{HasPref}(x)$. It will suffice to show that $\mathbf{HasNotPref}(x) \subseteq \{0,1,2\}^* - \mathbf{HasPref}(x) \subseteq \mathbf{HasNotPref}(x)$.

Suppose $w \in \mathbf{HasNotPref}(x)$. Then $w \in \{0,1,2\}^*$ and x is not a prefix of w. Suppose, toward a contradiction, that $w \in \mathbf{HasPref}(x)$. Then x is a prefix of w—contradiction. Thus $w \notin \mathbf{HasPref}(x)$, completing the proof that $w \in \{0,1,2\}^* - \mathbf{HasPref}(x)$.

Suppose $w \in \{0,1,2\}^* - \mathbf{HasPref}(x)$. Then $w \in \{0,1,2\}^*$ and $w \notin \mathbf{HasPref}(x)$. Suppose, toward a contradiction, that x is a prefix of w. Then $w \in \mathbf{HasPref}(x)$—contradiction. Hence x is not a prefix of w, completing the proof that $w \in \mathbf{HasNotPref}(x)$.

(5) Suppose $x, y, z \in \{0,1,2\}^*$. We must show that

$$\mathbf{NotSur}(x,y,z) = \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^* \cup \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z).$$

It will suffice to show that

$$\begin{aligned}\mathbf{NotSur}(x,y,z) &\subseteq \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^* \cup \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z)\\ &\subseteq \mathbf{NotSur}(x,y,z).\end{aligned}$$

Suppose $w \in \mathbf{NotSur}(x, y, z)$. Thus $w \in \{0,1,2\}^*$ and there are $u, v \in \{0,1,2\}^*$ such that $w = uyv$ and either x is not a suffix of u or z is not a prefix of v. There are two cases to consider.

- Suppose x is not a suffix of u. Thus $u \in \mathbf{HasNotSuf}(x)$, so that $w = uyv \in \mathbf{HasNotSuf}(x)\,\{y\}\,\{0,1,2\}^*$. Hence $w \in \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^* \cup \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z)$.
- Suppose z is not a prefix of v. Thus $v \in \mathbf{HasNotPref}(z)$, so that $w = uyv \in \{0,1,2\}^*\,\{y\}\,\mathbf{HasNotPref}(z)$. Hence $w \in \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^* \cup \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z)$.

Suppose $w \in \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^* \cup \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z)$. There are two cases to consider.

- Suppose $w \in \mathbf{HasNotSuf}(x)\{y\}\{0,1,2\}^*$. Then $w = uyv$ for some $u \in \mathbf{HasNotSuf}(x)$ and $v \in \{0,1,2\}^*$. Hence $u \in \{0,1,2\}^*$ and x is not a suffix of u, so that $w \in \{0,1,2\}^*$. Thus $w = uyv$ and either x is not a suffix of u or z is not a prefix of v, so that $w \in \mathbf{NotSur}(x, y, z)$.
- Suppose $w \in \{0,1,2\}^*\{y\}\mathbf{HasNotPref}(z)$. Then $w = uyv$ for some $u \in \{0,1,2\}^*$ and $v \in \mathbf{HasNotPref}(z)$. Hence $v \in \{0,1,2\}^*$ and z is not a prefix of v, so that $w \in \{0,1,2\}^*$. Thus $w = uyv$ and either x is not a suffix of u or z is not a prefix of v, so that $w \in \mathbf{NotSur}(x, y, z)$.

(6) Suppose $x, y, z \in \{0,1,2\}^*$. We must show that $\mathbf{Sur}(x, y, z) = \{0,1,2\}^* - \mathbf{NotSur}(x, y, z)$. It will suffice to show that $\mathbf{Sur}(x, y, z) \subseteq \{0,1,2\}^* - \mathbf{NotSur}(x, y, z) \subseteq \mathbf{Sur}(x, y, z)$.

Suppose $w \in \mathbf{Sur}(x, y, z)$. Then $w \in \{0,1,2\}^*$ and, (†) for all $u, v \in \{0,1,2\}^*$, if $w = uyv$, then x is a suffix of u and z is a prefix of v. Suppose, toward a contradiction, that $w \in \mathbf{NotSur}(x, y, z)$. Then there are $u, v \in \{0,1,2\}^*$ such that $w = uyv$ and either x is not a suffix of u or z is not a prefix of v. But this contradicts (†). Thus $w \notin \mathbf{NotSur}(x, y, z)$, completing the proof that $w \in \{0,1,2\}^* - \mathbf{NotSur}(x, y, z)$.

Suppose $w \in \{0,1,2\}^* - \mathbf{NotSur}(x, y, z)$. Thus $w \in \{0,1,2\}^*$ and $w \notin \mathbf{NotSur}(x, y, z)$. To see that $w \in \mathbf{Sur}(x, y, z)$, suppose $u, v \in \{0,1,2\}^*$ and $w = uyv$. We must show that x is a suffix of u and z is a prefix of v. Suppose, toward a contradiction, that x is not a suffix of u. Then $w = uyv$ and either x is not a suffix of u or z is not a prefix of v, so that $w \in \mathbf{NotSur}(x, y, z)$—contradiction. Thus x is a suffix of u. Suppose, toward a contradiction, that z is not a prefix of v. Then $w = uyv$ and either x is not a suffix of u or z is not a prefix of v, so that $w \in \mathbf{NotSur}(x, y, z)$—contradiction. Thus z is a prefix of v.

□

Next, we define some useful functions. Define $\mathbf{faToDFA} \in \mathbf{FA} \to \mathbf{DFA}$ by:

$$\mathbf{faToDFA} = \mathbf{nfaToDFA} \circ \mathbf{efaToNFA} \circ \mathbf{faToEFA}.$$

Then we have that, for all $M \in \mathbf{FA}$,

$$\begin{aligned} L(\mathbf{faToDFA}(M)) &= L(\mathbf{nfaToDFA}(\mathbf{efaToNFA}(\mathbf{faToEFA}(M)))) \\ &= L(\mathbf{efaToNFA}(\mathbf{faToEFA}(M))) = L(\mathbf{faToEFA}(M)) = L(M). \end{aligned}$$

Define $\mathbf{regToDFA} \in \mathbf{Reg} \to \mathbf{DFA}$ by:

$$\mathbf{regToDFA} = \mathbf{faToDFA} \circ \mathbf{regToFA}.$$

Then we have that, for all $\alpha \in \mathbf{Reg}$,

$$L(\mathbf{regToDFA}(\alpha)) = L(\mathbf{faToDFA}(\mathbf{regToFA}(\alpha))) = L(\mathbf{regToFA}(\alpha)) = L(\alpha).$$

Define $\mathbf{minAndRen} \in \mathbf{DFA} \to \mathbf{DFA}$ by: for all $M \in \mathbf{DFA}$,

$$\mathbf{minAndRen}(M) = \mathbf{renameStatesCanonically}(\mathbf{minimize}(M)).$$

Then, for all $M \in \mathbf{FA}$,

$$\begin{aligned} L(\mathbf{minAndRen}(M)) &= L(\mathbf{renameStatesCanonically}(\mathbf{minimize}(M))) \\ &= L(\mathbf{minimize}(M)) = L(M). \end{aligned}$$

Define the DFA **allStrDFA** by:

$$\mathbf{allStrDFA} = \mathbf{minAndRen}(\mathbf{regToDFA}((0 + 1 + 2)^*)).$$

Then, we have that

$$\begin{aligned} L(\mathbf{allStrDFA}) &= L(\mathbf{minAndRen}(\mathbf{regToDFA}((0 + 1 + 2)^*))) \\ &= L(\mathbf{regToDFA}((0 + 1 + 2)^*)) \\ &= L((0 + 1 + 2)^*) = \{0, 1, 2\}^*. \end{aligned}$$

Let the FA **allStrFA** be **allStrDFA**. Thus $L(\mathbf{allStrFA}) = L(\mathbf{allStrDFA}) = \{0, 1, 2\}^*$.

Define $\mathbf{hasSufFA} \in \{0, 1, 2\}^* \to \mathbf{FA}$ by: for all $x \in \{0, 1, 2\}^*$,

$$\mathbf{hasSufFA}(x) = \mathbf{concat}(\mathbf{allStrFA}, \mathbf{strToFA}(x)).$$

Then, we have that, for all $x \in \{0, 1, 2\}^*$,

$$\begin{aligned} L(\mathbf{hasSufFA}(x)) &= L(\mathbf{concat}(\mathbf{allStrFA}, \mathbf{strToFA}(x))) \\ &= L(\mathbf{allStrFA})\, L(\mathbf{strToFA}(x)) \\ &= \{0, 1, 2\}^* \{x\} \\ &= \mathbf{HasSuf}(x), \end{aligned}$$

by Lemma ES5.3.1(1).

Define $\mathbf{hasSufDFA} \in \{0, 1, 2\}^* \to \mathbf{DFA}$ by: for all $x \in \{0, 1, 2\}^*$,

$$\mathbf{hasSufDFA}(x) = \mathbf{minAndRen}(\mathbf{faToDFA}(\mathbf{hasSufFA}(x))).$$

Then, we have that, for all $x \in \{0, 1, 2\}^*$,

$$\begin{aligned} L(\mathbf{hasSufDFA}(x)) &= L(\mathbf{minAndRen}(\mathbf{faToDFA}(\mathbf{hasSufFA}(x)))) \\ &= L(\mathbf{faToDFA}(\mathbf{hasSufFA}(x))) \\ &= L(\mathbf{hasSufFA}(x)) \\ &= \mathbf{HasSuf}(x). \end{aligned}$$

Define $\mathbf{hasNotSufDFA} \in \{0,1,2\}^* \to \mathbf{DFA}$ by: for all $x \in \{0,1,2\}^*$,

$$\mathbf{hasNotSufDFA}(x) = \mathbf{minus}(\mathbf{allStrDFA}, \mathbf{hasSufDFA}(x)).$$

Then, we have that, for all $x \in \{0,1,2\}^*$,

$$\begin{aligned} L(\mathbf{hasNotSufDFA}(x)) &= L(\mathbf{minus}(\mathbf{allStrDFA}, \mathbf{hasSufDFA}(x))) \\ &= L(\mathbf{allStrDFA}) - L(\mathbf{hasSufDFA}(x)) \\ &= \{0,1,2\}^* - \mathbf{HasSuf}(x) \\ &= \mathbf{HasNotSuf}(x), \end{aligned}$$

by Lemma ES5.3.1(2).

Define $\mathbf{hasNotSufFA} \in \{0,1,2\}^* \to \mathbf{FA}$ by: for all $x \in \{0,1,2\}^*$, $\mathbf{hasNotSufFA}(x) = \mathbf{hasNotSufDFA}(x)$. Then, for all $x \in \{0,1,2\}^*$,

$$L(\mathbf{hasNotSufFA}(x)) = L(\mathbf{hasNotSufDFA}(x)) = \mathbf{HasNotSuf}(x).$$

Define $\mathbf{hasPrefFA} \in \{0,1,2\}^* \to \mathbf{FA}$ by: for all $x \in \{0,1,2\}^*$,

$$\mathbf{hasPrefFA}(x) = \mathbf{concat}(\mathbf{strToFA}(x), \mathbf{allStrFA}).$$

Then, we have that, for all $x \in \{0,1,2\}^*$,

$$\begin{aligned} L(\mathbf{hasPrefFA}(x)) &= L(\mathbf{concat}(\mathbf{strToFA}(x), \mathbf{allStrFA})) \\ &= L(\mathbf{strToFA}(x))\, L(\mathbf{allStrFA}) \\ &= \{x\}\{0,1,2\}^* \\ &= \mathbf{HasPref}(x), \end{aligned}$$

by Lemma ES5.3.1(3).

Define $\mathbf{hasPrefDFA} \in \{0,1,2\}^* \to \mathbf{DFA}$ by: for all $x \in \{0,1,2\}^*$,

$$\mathbf{hasPrefDFA}(x) = \mathbf{minAndRen}(\mathbf{faToDFA}(\mathbf{hasPrefFA}(x))).$$

Then, we have that, for all $x \in \{0,1,2\}^*$,

$$\begin{aligned} L(\mathbf{hasPrefDFA}(x)) &= L(\mathbf{minAndRen}(\mathbf{faToDFA}(\mathbf{hasPrefFA}(x)))) \\ &= L(\mathbf{faToDFA}(\mathbf{hasPrefFA}(x))) \\ &= L(\mathbf{hasPrefFA}(x)) \\ &= \mathbf{HasPref}(x). \end{aligned}$$

Define $\mathbf{hasNotPrefDFA} \in \{0,1,2\}^* \to \mathbf{DFA}$ by: for all $x \in \{0,1,2\}^*$,

$$\mathbf{hasNotPrefDFA}(x) = \mathbf{minus}(\mathbf{allStrDFA}, \mathbf{hasPrefDFA}(x)).$$

Then, we have that, for all $x \in \{0,1,2\}^*$,

$$\begin{aligned} L(\mathbf{hasNotPrefDFA}(x)) &= L(\mathbf{minus}(\mathbf{allStrDFA}, \mathbf{hasPrefDFA}(x))) \\ &= L(\mathbf{allStrDFA}) - L(\mathbf{hasPrefDFA}(x)) \\ &= \{0,1,2\}^* - \mathbf{HasPref}(x) \\ &= \mathbf{HasNotPref}(x), \end{aligned}$$

by Lemma ES5.3.1(4).

Define $\mathbf{hasNotPrefFA} \in \{0,1,2\}^* \to \mathbf{FA}$ by: for all $x \in \{0,1,2\}^*$, $\mathbf{hasNotPrefFA}(x) = \mathbf{hasNotPrefDFA}(x)$. Then, for all $x \in \{0,1,2\}^*$,

$$L(\mathbf{hasNotPrefFA}(x)) = L(\mathbf{hasNotPrefDFA}(x)) = \mathbf{HasNotPref}(x).$$

Define $\mathbf{notSurFA} \in \{0,1,2\}^* \times \{0,1,2\}^* \times \{0,1,2\}^* \to \mathbf{FA}$ by: for all $x, y, z \in \{0,1,2\}^*$,

$$\begin{aligned}\mathbf{notSurFA}(x,y,z) = \mathbf{union}(&\mathbf{concat}(\mathbf{hasNotSufFA}(x), \mathbf{concat}(\mathbf{strToFA}(y), \mathbf{allStrFA})),\\ &\mathbf{concat}(\mathbf{allStrFA}, \mathbf{concat}(\mathbf{strToFA}(y), \mathbf{hasNotPrefFA}(z)))).\end{aligned}$$

Then, we have that, for all $x, y, z \in \{0,1,2\}^*$,

$$\begin{aligned}L(\mathbf{notSurFA}(x,y,z)) &= L(\mathbf{union}(\mathbf{concat}(\mathbf{hasNotSufFA}(x), \mathbf{concat}(\mathbf{strToFA}(y), \mathbf{allStrFA})),\\ &\qquad \mathbf{concat}(\mathbf{allStrFA}, \mathbf{concat}(\mathbf{strToFA}(y), \mathbf{hasNotPrefFA}(z)))))\\ &= L(\mathbf{concat}(\mathbf{hasNotSufFA}(x), \mathbf{concat}(\mathbf{strToFA}(y), \mathbf{allStrFA}))) \cup\\ &\quad\; L(\mathbf{concat}(\mathbf{allStrFA}, \mathbf{concat}(\mathbf{strToFA}(y), \mathbf{hasNotPrefFA}(z))))\\ &= L(\mathbf{hasNotSufFA}(x))\, L(\mathbf{strToFA}(y))\, L(\mathbf{allStrFA}) \cup\\ &\quad\; L(\mathbf{allStrFA})\, L(\mathbf{strToFA}(y))\, L(\mathbf{hasNotPrefFA}(z))\\ &= \mathbf{HasNotSuf}(x)\, \{y\}\, \{0,1,2\}^* \cup \{0,1,2\}^*\, \{y\}\, \mathbf{HasNotPref}(z)\\ &= \mathbf{NotSur}(x,y,z),\end{aligned}$$

by Lemma ES5.3.1(5).

Define $\mathbf{notSurDFA} \in \{0,1,2\}^* \times \{0,1,2\}^* \times \{0,1,2\}^* \to \mathbf{DFA}$ by: for all $x, y, z \in \{0,1,2\}^*$,

$$\mathbf{notSurDFA}(x,y,z) = \mathbf{minAndRen}(\mathbf{faToDFA}(\mathbf{notSurFA}(x,y,z))).$$

Then, we have that, for all $x, y, z \in \{0,1,2\}^*$,

$$\begin{aligned}L(\mathbf{notSurDFA}(x,y,z)) &= L(\mathbf{minAndRen}(\mathbf{faToDFA}(\mathbf{notSurFA}(x,y,z))))\\ &= L(\mathbf{faToDFA}(\mathbf{notSurFA}(x,y,z)))\\ &= L(\mathbf{notSurFA}(x,y,z))\\ &= \mathbf{NotSur}(x,y,z).\end{aligned}$$

Finally define $\mathbf{surDFA} \in \{0,1,2\}^* \times \{0,1,2\}^* \times \{0,1,2\}^* \to \mathbf{DFA}$ by: for all $x, y, z \in \{0,1,2\}^*$,

$$\mathbf{surDFA}(x,y,z) = \mathbf{minAndRen}(\mathbf{minus}(\mathbf{allStrDFA}, \mathbf{notSurDFA}(x,y,z))).$$

Then we have that, for all $x, y, z \in \{0,1,2\}^*$,

$$\begin{aligned}L(\mathbf{surDFA}(x,y,z)) &= L(\mathbf{minAndRen}(\mathbf{minus}(\mathbf{allStrDFA}, \mathbf{notSurDFA}(x,y,z))))\\ &= L(\mathbf{minus}(\mathbf{allStrDFA}, \mathbf{notSurDFA}(x,y,z)))\\ &= L(\mathbf{allStrDFA}) - L(\mathbf{notSurDFA}(x,y,z))\\ &= \{0,1,2\}^* - \mathbf{NotSur}(x,y,z)\\ &= \mathbf{Sur}(x,y,z),\end{aligned}$$

by Lemma ES5.3.1(6), and **surDFA**(x, y, z) has as few states as possible, because the last step in its definition is **minAndRen**.

(b) First, we put the text

```
val faToDFA   = nfaToDFA o efaToNFA o faToEFA;
val regToDFA  = faToDFA o regToFA;
val minAndRen = DFA.renameStatesCanonically o DFA.minimize;
val allStrDFA = minAndRen(regToDFA(Reg.fromString "(0 + 1 + 2)*"));
val allStrFA  = injDFAToFA allStrDFA;

fun hasSufFA x = FA.concat(allStrFA, strToFA x);
val hasSufDFA = minAndRen o faToDFA o hasSufFA;
fun hasNotSufDFA x = DFA.minus(allStrDFA, hasSufDFA x);
val hasNotSufFA = injDFAToFA o hasNotSufDFA;

fun hasPrefFA x = FA.concat(strToFA x, allStrFA);
val hasPrefDFA = minAndRen o faToDFA o hasPrefFA;
fun hasNotPrefDFA x = DFA.minus(allStrDFA, hasPrefDFA x);
val hasNotPrefFA = injDFAToFA o hasNotPrefDFA;

fun notSurFA(x,y,z) =
      FA.union(FA.concat(hasNotSufFA x,
                         FA.concat(strToFA y,
                                   allStrFA)),
               FA.concat(allStrFA,
                         FA.concat(strToFA y,
                                   hasNotPrefFA z)));
val notSurDFA = minAndRen o faToDFA o notSurFA;

fun surDFA(x, y, z) = minAndRen(DFA.minus(allStrDFA, notSurDFA(x, y, z)));
```

in the file `sur.sml`. Then we invoke Forlan and proceed as follows:

```
- use "sur.sml";
[opening sur.sml]
val faToDFA = fn : fa -> dfa
val regToDFA = fn : reg -> dfa
val minAndRen = fn : dfa -> dfa
val allStrDFA = - : dfa
val allStrFA = - : fa
val hasSufFA = fn : str -> fa
val hasSufDFA = fn : str -> dfa
val hasNotSufDFA = fn : str -> dfa
val hasNotSufFA = fn : str -> fa
val hasPrefFA = fn : str -> fa
val hasPrefDFA = fn : str -> dfa
val hasNotPrefDFA = fn : str -> dfa
val hasNotPrefFA = fn : str -> fa
```

```
val notSurFA = fn : str * str * str -> fa
val notSurDFA = fn : str * str * str -> dfa
val surDFA = fn : str * str * str -> dfa
val it = () : unit
- val dfa =
=       surDFA(Str.fromString "00",
=              Str.fromString "11",
=              Str.fromString "22");
val dfa = - : dfa
- DFA.output("", dfa);
{states}
A, B, C, D, E, F, G, H
{start state}
C
{accepting states}
A, B, C, E, F
{transitions}
A, 0 -> A; A, 1 -> E; A, 2 -> C; B, 0 -> A; B, 1 -> F; B, 2 -> C; C, 0 -> B;
C, 1 -> F; C, 2 -> C; D, 0 -> H; D, 1 -> H; D, 2 -> C; E, 0 -> B; E, 1 -> G;
E, 2 -> C; F, 0 -> B; F, 1 -> H; F, 2 -> C; G, 0 -> H; G, 1 -> H; G, 2 -> D;
H, 0 -> H; H, 1 -> H; H, 2 -> H
val it = () : unit
```

Here is a drawing of `dfa`: