CIS 570 — Introduction to Formal Language Theory — Fall 2008

Final Examination

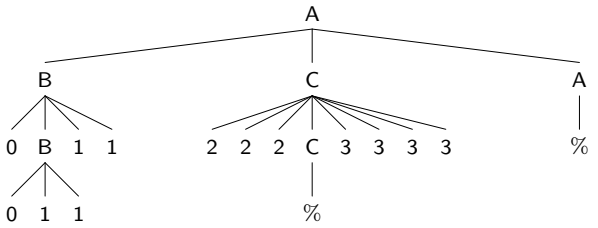
Model Answers

Question 1

(a)

$$\begin{aligned}
&\mathsf{A} \to \% \mid \mathsf{BCA},\\
&\mathsf{B} \to 011 \mid 0\mathsf{B}11,\\
&\mathsf{C} \to \% \mid 222\mathsf{C}3333.
\end{aligned}$$

(b) Let pt_1 be:



We have that $\mathbf{yield}\, pt_1 = 0011112223333$.

Question 2

(a) First, we carry out three standard declarations:

```
- val faToDFA   = nfaToDFA o efaToNFA o faToEFA;
val faToDFA = fn : fa -> dfa
- val regToDFA  = faToDFA o regToFA;
val regToDFA = fn : reg -> dfa
- val minAndRen = DFA.renameStatesCanonically o DFA.minimize;
val minAndRen = fn : dfa -> dfa
```

Next, we declare `allStrDFA` to be a DFA accepting $\{0,1\}^*$:

```
- val allStrDFA = minAndRen(regToDFA(Reg.fromString "(0 + 1)*"));
val allStrDFA = - : dfa
```

Next, we declare DFAs `evenLenDFA`, `has000DFA` and `has111DFA` accepting $\{\, w \in \{0,1\}^* \mid |w| \text{ is even} \,\}$, $\{\, w \in \{0,1\}^* \mid 000 \text{ is a substring of } w \,\}$ and $\{\, w \in \{0,1\}^* \mid 111 \text{ is a substring of } w \,\}$, respectively:

```
- val evenLenDFA = minAndRen(regToDFA(Reg.fromString "((0 + 1)(0 + 1))*"));
val evenLenDFA = - : dfa
```

```
- val has000DFA  = minAndRen(regToDFA(Reg.fromString "(0 + 1)*000(0 + 1)*"));
val has000DFA = - : dfa
- val has111DFA  = minAndRen(regToDFA(Reg.fromString "(0 + 1)*111(0 + 1)*"));
val has111DFA = - : dfa
```

Next, we declare a minimized DFA `validDFA` accepting X:

```
- val validDFA =
=       minAndRen(DFA.inter(evenLenDFA,
=                           DFA.inter(has000DFA,
=                                     has111DFA)));
val validDFA = - : dfa
```

(b) Continuing our Forlan session, we declare `allNEStrDFA` to be a DFA accepting $\{\, w \in \{0,1\}^* \mid w \neq \% \,\}$:

```
- val allNEStrDFA = minAndRen(regToDFA(Reg.fromString "(0 + 1)(0 + 1)*"));
val allNEStrDFA = - : dfa
```

Next, we declare an FA `hasValidPropSubFA` accepting

$$\{\, w \in \{0,1\}^* \mid \text{there is a proper substring } v \text{ of } w \text{ such that } v \in X \,\},$$

and then declare a minimized DFA `hasValidPropSubDFA` accepting this language:

```
- val hasValidPropSubFA =
=       FA.union(FA.concat(injDFAToFA allNEStrDFA,
=                          injDFAToFA validDFA),
=                FA.union(FA.concat(injDFAToFA validDFA,
=                                   injDFAToFA allNEStrDFA),
=                         FA.concat(injDFAToFA allNEStrDFA,
=                                   FA.concat(injDFAToFA validDFA,
=                                             injDFAToFA allNEStrDFA))));
val hasValidPropSubFA = - : fa
- val hasValidPropSubDFA = minAndRen(faToDFA hasValidPropSubFA);
val hasValidPropSubDFA = - : dfa
```

Next, we declare a DFA `hasNoValidPropSubDFA` accepting

$$\{\, w \in \{0,1\}^* \mid \text{there is no proper substring } v \text{ of } w \text{ such that } v \in X \,\} :$$

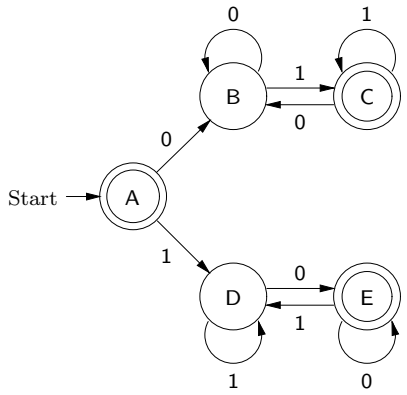
```
- val hasNoValidPropSubDFA = DFA.minus(allStrDFA, hasValidPropSubDFA);
val hasNoValidPropSubDFA = - : dfa
```

Finally, we declare a minimized DFA `onlyValidDFA` accepting Y:

```
- val onlyValidDFA = minAndRen(DFA.inter(validDFA, hasNoValidPropSubDFA));
val onlyValidDFA = - : dfa
```

Question 3

(a)



(b) First, we note that, for all $w \in \{0,1\}^*$:

- $w \in X$ iff $\mathsf{blocks}\, w$ is even;
- $w \notin X$ iff $\mathsf{blocks}\, w$ is odd.

Next, we use induction on Λ to prove that:

(A) For all $w \in \Lambda_{\mathsf{A}}$, $w = \%$.

(B) For all $w \in \Lambda_{\mathsf{B}}$, $\mathsf{blocks}\, w$ is odd and 0 is a suffix of w.

(C) For all $w \in \Lambda_{\mathsf{C}}$, $\mathsf{blocks}\, w$ is even and 1 is a suffix of w.

(D) For all $w \in \Lambda_{\mathsf{D}}$, $\mathsf{blocks}\, w$ is odd and 1 is a suffix of w.

(E) For all $w \in \Lambda_{\mathsf{E}}$, $\mathsf{blocks}\, w$ is even and 0 is a suffix of w.

There are 11 steps to show.

- (empty string) We have that $\% = \%$.
- $(\mathsf{A}, 0 \to \mathsf{B})$ Suppose $w \in \Lambda_{\mathsf{A}}$ (so that $w \in (\textbf{alphabet}\, M)^* = \{0,1\}^*$), and assume the inductive hypothesis, $w = \%$. Thus $\mathsf{blocks}(w0) = 1$ is odd, and 0 is a suffix of $w0$.
- $(\mathsf{A}, 1 \to \mathsf{D})$ Suppose $w \in \Lambda_{\mathsf{A}}$, and assume the inductive hypothesis, $w = \%$. Thus $\mathsf{blocks}(w1) = 1$ is odd, and 1 is a suffix of $w1$.
- $(\mathsf{B}, 0 \to \mathsf{B})$ Suppose $w \in \Lambda_{\mathsf{B}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is odd and 0 is a suffix of w. Thus $\mathsf{blocks}(w0) = \mathsf{blocks}\, w$ is odd and 0 is a suffix of $w0$.
- $(\mathsf{B}, 1 \to \mathsf{C})$ Suppose $w \in \Lambda_{\mathsf{B}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is odd and 0 is a suffix of w. Thus $\mathsf{blocks}(w1) = \mathsf{blocks}\, w + 1$ is even and 1 is a suffix of $w1$.

- $(\mathsf{C}, 0 \to \mathsf{B})$ Suppose $w \in \Lambda_{\mathsf{C}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is even and 1 is a suffix of w. Thus $\mathsf{blocks}(w0) = \mathsf{blocks}\, w + 1$ is odd and 0 is a suffix of $w0$.
- $(\mathsf{C}, 1 \to \mathsf{C})$ Suppose $w \in \Lambda_{\mathsf{C}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is even and 1 is a suffix of w. Thus $\mathsf{blocks}(w1) = \mathsf{blocks}\, w$ is even and 1 is a suffix of $w1$.
- $(\mathsf{D}, 0 \to \mathsf{E})$ Suppose $w \in \Lambda_{\mathsf{D}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is odd and 1 is a suffix of w. Thus $\mathsf{blocks}(w0) = \mathsf{blocks}\, w + 1$ is even and 0 is a suffix of $w0$.
- $(\mathsf{D}, 1 \to \mathsf{D})$ Suppose $w \in \Lambda_{\mathsf{D}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is odd and 1 is a suffix of w. Thus $\mathsf{blocks}(w1) = \mathsf{blocks}\, w$ is odd and 1 is a suffix of $w1$.
- $(\mathsf{E}, 0 \to \mathsf{E})$ Suppose $w \in \Lambda_{\mathsf{E}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is even and 0 is a suffix of w. Thus $\mathsf{blocks}(w0) = \mathsf{blocks}\, w$ is even and 0 is a suffix of $w0$.
- $(\mathsf{E}, 1 \to \mathsf{D})$ Suppose $w \in \Lambda_{\mathsf{E}}$, and assume the inductive hypothesis, $\mathsf{blocks}\, w$ is even and 0 is a suffix of w. Thus $\mathsf{blocks}(w1) = \mathsf{blocks}\, w + 1$ is odd and 1 is a suffix of $w1$.

Now, we use the result of our induction on Λ to prove that $L(M) = X$.

- $(L(M) \subseteq X)$ Suppose $w \in L(M)$. Hence $w \in (\mathbf{alphabet}\, M)^* = \{0, 1\}^*$ and $w \in L(M) = \Lambda_{\mathsf{A}} \cup \Lambda_{\mathsf{C}} \cup \Lambda_{\mathsf{E}}$. Thus, there are three cases to consider.
  - Suppose $w \in \Lambda_{\mathsf{A}}$. By Part (A), we have that $w = \%$. Because $\mathsf{blocks}\, w = \mathsf{blocks}\, \% = 0$ is even, we have that $w \in X$.
  - Suppose $w \in \Lambda_{\mathsf{C}}$. By Part (C), we have that $\mathsf{blocks}\, w$ is even, so that $w \in X$.
  - Suppose $w \in \Lambda_{\mathsf{E}}$. By Part (E), we have that $\mathsf{blocks}\, w$ is even, so that $w \in X$.
- $(X \subseteq L(M))$ Suppose $w \in X$. Since $X \subseteq \{0, 1\}^*$, we have that $w \in \{0, 1\}^*$. Suppose, toward a contradiction, that $w \notin L(M)$. Thus $w \notin L(M) = \Lambda_{\mathsf{A}} \cup \Lambda_{\mathsf{C}} \cup \Lambda_{\mathsf{E}}$. But $w \in \{0, 1\}^* = (\mathbf{alphabet}\, M)^* = \Lambda_{\mathsf{A}} \cup \Lambda_{\mathsf{B}} \cup \Lambda_{\mathsf{C}} \cup \Lambda_{\mathsf{D}} \cup \Lambda_{\mathsf{E}}$, so that $w \in \Lambda_{\mathsf{B}} \cup \Lambda_{\mathsf{D}}$. Thus, there are two cases to consider.
  - Suppose $w \in \Lambda_{\mathsf{B}}$. By Part (B), we have that $\mathsf{blocks}\, w$ is odd, so that $w \notin X$—contradiction.
  - Suppose $w \in \Lambda_{\mathsf{D}}$. By Part (D), we have that $\mathsf{blocks}\, w$ is odd, so that $w \notin X$—contradiction.

  Since we obtained a contradiction in both cases, we have an overall contradiction. Thus $w \in L(M)$.