CS 516—Software Foundations via Formal Languages—Spring 2022

Problem Set 5

Due by 5pm on Friday, April 8
Submission via Gradescope and GitHub

Problem 1 (16 points)

Define a function $\mathbf{diff} \in \{\mathsf{0}, \mathsf{1}\}^* \to \mathbb{Z}$ by: for all $w \in \{\mathsf{0}, \mathsf{1}\}^*$,

$$\mathbf{diff}\, w = \text{the number of } \mathsf{1}\text{'s in } w - \text{the number of } \mathsf{0}\text{'s in } w.$$

Let

$$X = \{\, w \in \{\mathsf{0}, \mathsf{1}\}^* \mid \text{for all substrings } v \text{ of } w,\ -2 \leq \mathbf{diff}\, v \leq 2 \,\}.$$

Use Forlan to find a regular expression α such that $L(\alpha) = X$, trying to make α be as simple as possible, and trying to do as much as possible of the work of finding α using Forlan. You may assume the results about X that were proved in the model answers to the old and new Problem Set 4s. Include a transcript of your Forlan session.

Problem 2 (18 points)

Define $\mathbf{Subst} \in \mathbf{Lan} \times \mathbf{Str} \times \mathbf{Str} \to \mathbf{Lan}$ by:

$$\mathbf{Subst}(L, x, y) = \begin{cases} (L - \{x\}) \cup \{y\} & \text{if } x \in L, \\ L & \text{if } x \notin L. \end{cases}$$

For example: $\mathbf{Subst}(\{\mathsf{01}, \mathsf{10}\}, \mathsf{01}, \mathsf{11}) = \{\mathsf{11}, \mathsf{10}\}$; $\mathbf{Subst}(\{\mathsf{01}, \mathsf{10}\}, \mathsf{01}, \mathsf{01}) = \{\mathsf{01}, \mathsf{10}\}$; $\mathbf{Subst}(\{\mathsf{01}, \mathsf{10}\}, \mathsf{01}, \mathsf{10}) = \{\mathsf{10}\}$; and $\mathbf{Subst}(\{\mathsf{01}, \mathsf{10}\}, \mathsf{11}, \mathsf{12}) = \{\mathsf{01}, \mathsf{10}\}$.

In a file `ps5-p2.sml`, define a Forlan/SML function

```
val subst : fa * str * str -> fa
```

such that `subst`(M, x, y) returns an FA N such that $L(N) = \mathbf{Subst}(L(M), x, y)$. Use Forlan to test your function, and include a transcript of your Forlan session. (In Section 3.13, we will learn a method for testing the equivalence of FAs. You may use this method, or may do your testing in another way.)

You should put your `ps5-p2.sml` in the subdirectory `CS516-PS5` of your private GitHub repository. If you define any functions as part of your testing, they should be included in a file `ps5-p2-testing.sml` in this directory.

Problem 3 (16 points)

We say that a string w is *super-accepted* by a finite automaton M iff

$$\emptyset \neq \Delta_M(\{s_M\}, w) \subseteq A_M.$$

It is easy to see that, if w is super-accepted by M, then w is accepted by M.

In a file `ps5-p3.sml`, define a Forlan/SML function

```
val superAccepted : fa -> str -> bool
```

such that `superAccepted` $M\,w$ tests whether w is super-accepted by M. Consider the function

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

of type `reg -> str -> bool`. Use Forlan to show there are inputs to `test` that cause it to return `false`. Include a transcript of your Forlan session.

You should put your `ps5-p3.sml` in the subdirectory `CS516-PS5` of your private GitHub repository.

Problem 4 (50 points)

Define $f \in \{0,1\}^* \to \mathcal{P}(\{0,1\}^*)$ by:

$$f\,w = \{\, y \in \{0,1\}^* \mid y000 \text{ is a prefix of } w \,\}.$$

Since $f\,w$ is always finite, we can define $g \in \{0,1\}^* \to \mathbb{N}$ by: $g\,w = |f\,w|$. Let $X = \{\, w \in \{0,1\}^* \mid g\,w \text{ is even} \,\}$.

For example:

- $f(0001000) = \{\%, 0001\}$, so that $g(0001000) = |\{\%, 0001\}| = 2$, and thus $0001000 \in X$; and
- $f(00001000) = \{\%, 0, 00001\}$, so that $g(00001000) = |\{\%, 0, 00001\}| = 3$, and thus $00001000 \notin X$.

(Informally, $g\,w$ is the *number of occurrences of* 000 *in* w. But in part (b), you must work in terms of the definitions of f and g, proving whatever properties you need, in a lemma or lemmas.)

(a) Find a DFA M such that $L(M) = X$. (Hint: although *not* required, you can use Forlan to test your M, before attempting part (b)). [10 points]

(b) Prove that $L(M) = X$. [40 points]