CIS 570 — Introduction to Formal Language Theory — Fall 2007

Exercise Set 2

Due by 4:00 p.m. on Tuesday, October 9

Exercise 1 (20 points)

Let $X = \{ w \in \{0,1\}^* \mid |w| \leq 4 \text{ and neither } 000 \text{ nor } 111 \text{ is a substring of } w \}$. Use Forlan to find and show the correctness of a regular expression α such that $L(\alpha) = X$. Try to minimize the size of α, and use Forlan to display the size of α. Try to do as much as possible of the work of finding and showing the correctness of α using Forlan. (Include a listing of your Forlan session.)

Exercise 2 (20 points)

(a) Prove that, for all $n \in \mathbb{N}$ and $A, B \in \mathbf{Lan}$, if $n \geq 1$ and $A^n \subseteq B$, then $A^{n+1}A^* \cup B = A^n A^* \cup B$. [15 points]

(b) Prove that, for all $n \in \mathbb{N}$ and $\alpha, \beta \in \mathbf{Reg}$, if $n \geq 1$ and $L(\alpha^n) \subseteq L(\beta)$, then $\alpha^{n+1}\alpha^* + \beta \approx \alpha^n\alpha^* + \beta$. [5 points]

Exercise 3 (35 points)

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

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

Thus:

- $\mathbf{diff}(\%) = 0$;
- $\mathbf{diff}(0) = -1$;
- $\mathbf{diff}(1) = 1$;
- for all $x, y \in \{0,1\}^*$, $\mathbf{diff}(xy) = \mathbf{diff}(x) + \mathbf{diff}(y)$.

Let $X = \{ w \in \{0,1\}^* \mid \text{for all prefixes } v \text{ of } w, 0 \leq \mathbf{diff}(v) \leq 2 \}$.

(a) Find a regular expression α such that $L(\alpha) = X$. [10 points]

(b) Prove that your answer to Part (a) is correct. [25 points]

Exercise 4 (25 points)

Consider the model answer to Exercise 4(b) of Exercise Set 1. Turn the proof that $Y \subseteq X$ into the definition of a Standard ML/Forlan function

```
val expl : int * str -> unit
```

that, given an indentation level and an element $w \in Y$, prints out, at that indentation level, an explanation of why $w \in X$.

As closely as possible, make the structure of your function definition match the structure of the proof. In particular: induction in the proof should correspond to recursion in your function definition; division into cases in the proof should correspond to the use of conditionals/pattern matching in the function definition; and the use of the lemmas in the proof should correspond to the use of auxiliary functions in the function definition.

Then, use your function to finish writing a program whose main function

```
val explain : unit -> unit
```

reads (using `Str.input`) a string w from the standard input, issues an error message, if $w \notin Y$, and uses `expl` to explain why $w \in X$, otherwise.

For example:

- if your program is given the string 100011101, then it should output the following explanation for why this string is in X:

```
100011101 = 1 @ 0 @ 0011101 is in X, by (2)
  1 is in X, by (1)
  0011101 = 0 @ 011 @ 101 is in X, by (3)
    011 = 0 @ 1 @ 1 is in X, by (3)
      1 is in X, by (1)
      1 is in X, by (1)
    101 = 1 @ 0 @ 1 is in X, by (2)
      1 is in X, by (1)
      1 is in X, by (1)
```

- if your program is given the string 0110, then it should explain why this string isn't in Y, saying:

```
diff of string is 0 not 1
```

- if your program is given the string 01110, then it should explain why this string isn't in Y, saying:

```
prefix 0111 of string has diff 2 which is greater-than 1
```

- if your program is given the string 1020, then it should explain why this string isn't in Y, saying:

```
string has symbol other than 0/1 : 2
```

Supply a listing of your program, as well as a transcript showing how you tested your program. Keep an electronic copy of your program, in case you are asked to make it available for further testing.

Hint: you may adapt the SML/Forlan program solving Exercise 4 of last year's Exercise Set 2. The Forlan WWW site contains links to documentation on Standard ML and a useful comparison of OCaml and Standard ML.