CS 516—Software Foundations via Formal Languages—Spring 2022

Problem Set 2

Model Answers

Problem 1

Part (a)

It will suffice to use induction on X to show that, for all $w \in X$, $w \in Y$. There are five steps to show.

(1) We must show that $\% \in Y$, and this follows since $\% \in \{0,1\}^*$ and $\mathbf{diff}\,\% = 0$.

(2) Suppose $x, y \in X$, and assume the inductive hypothesis: $x, y \in Y$. We must show that $0x0y1 \in Y$. Because $x, y \in Y$, it follows that $0x0y1 \in \{0,1\}^*$. And, since $x, y \in Y$, we have that $\mathbf{diff}\,x = 0 = \mathbf{diff}\,y$, so that $\mathbf{diff}(0x0y1) = \mathbf{diff}\,0 + \mathbf{diff}\,x + \mathbf{diff}\,0 + \mathbf{diff}\,y + \mathbf{diff}\,1 = 1 + 0 + 1 + 0 + -2 = 0$, completing the proof that $0x0y1 \in Y$.

(3) Suppose $x, y \in X$, and assume the inductive hypothesis: $x, y \in Y$. We must show that $0x1y0 \in Y$. Because $x, y \in Y$, it follows that $0x1y0 \in \{0,1\}^*$. And, since $x, y \in Y$, we have that $\mathbf{diff}\,x = 0 = \mathbf{diff}\,y$, so that $\mathbf{diff}(0x1y0) = \mathbf{diff}\,0 + \mathbf{diff}\,x + \mathbf{diff}\,1 + \mathbf{diff}\,y + \mathbf{diff}\,0 = 1 + 0 + -2 + 0 + 1 = 0$, completing the proof that $0x1y0 \in Y$.

(4) Suppose $x, y \in X$, and assume the inductive hypothesis: $x, y \in Y$. We must show that $1x0y0 \in Y$. Because $x, y \in Y$, it follows that $1x0y0 \in \{0,1\}^*$. And, since $x, y \in Y$, we have that $\mathbf{diff}\,x = 0 = \mathbf{diff}\,y$, so that $\mathbf{diff}(1x0y0) = \mathbf{diff}\,1 + \mathbf{diff}\,x + \mathbf{diff}\,0 + \mathbf{diff}\,y + \mathbf{diff}\,0 = -2 + 0 + 1 + 0 + 1 = 0$, completing the proof that $1x0y0 \in Y$.

(5) Suppose $x, y \in X$, and assume the inductive hypothesis: $x, y \in Y$. We must show that $xy \in Y$. Because $x, y \in Y$, it follows that $xy \in \{0,1\}^*$. And, since $x, y \in Y$, we have that $\mathbf{diff}\,x = 0 = \mathbf{diff}\,y$, so that $\mathbf{diff}(xy) = \mathbf{diff}\,x + \mathbf{diff}\,y = 0 + 0 = 0$, completing the proof that $xy \in Y$.

Part (b)

We begin by proving a useful lemma:

Lemma PS2.1.1
For all $w \in \{0,1\}^*$, if $\mathbf{diff}\,w \geq 1$, then $w = x0y$, for some $x, y \in \{0,1\}^*$ such that $\mathbf{diff}\,x = 0$ and $\mathbf{diff}\,y = \mathbf{diff}\,w - 1$.

Proof. Suppose $w \in \{0,1\}^*$ and $\mathbf{diff}\,w \geq 1$. Let $u \in \{0,1\}^*$ be the shortest prefix of w such that $\mathbf{diff}\,u \geq 1$, and let $y \in \{0,1\}^*$ be such that $w = uy$. Then $u \neq \%$, so that $u = xb$ for some $x \in \{0,1\}^*$ and $b \in \{0,1\}$. Thus $w = uy = xby$. Since x is a shorter prefix of w than u, we have that $\mathbf{diff}\,x \leq 0$.

Suppose, toward a contradiction, that $b = \mathtt{1}$. Then $\textbf{diff}\, x + -2 = \textbf{diff}(x\mathtt{1}) = \textbf{diff}(xb) = \textbf{diff}\, u \geq 1$, so that $\textbf{diff}\, x \geq 3$—contradiction. Thus $b = \mathtt{0}$.

Summarizing, we have that $u = xb = x\mathtt{0}$, $w = uy = x\mathtt{0}y$, $\textbf{diff}\, u \geq 1$, $\textbf{diff}\, w \geq 1$ and $\textbf{diff}\, x \leq 0$. Since $\textbf{diff}\, x + 1 = \textbf{diff}(x\mathtt{0}) = \textbf{diff}\, u \geq 1$, we have that $\textbf{diff}\, x \geq 0$. But $\textbf{diff}\, x \leq 0$, and thus $\textbf{diff}\, x = 0$. Finally, since $\textbf{diff}\, w = \textbf{diff}(x\mathtt{0}y) = 0 + 1 + \textbf{diff}\, y = 1 + \textbf{diff}\, y$, we have that $\textbf{diff}\, y = \textbf{diff}\, w - 1$. □

Now, we use the lemma to prove that $Y \subseteq X$. Since $Y \subseteq \{\mathtt{0}, \mathtt{1}\}^*$, it will suffice to show that, for all $w \in \{\mathtt{0}, \mathtt{1}\}^*$,

$$\text{if } w \in Y, \text{ then } w \in X.$$

We proceed by strong string induction. Suppose $w \in \{\mathtt{0}, \mathtt{1}\}^*$, and assume the inductive hypothesis: for all $x \in \{\mathtt{0}, \mathtt{1}\}^*$, if x is a proper substring of w, then

$$\text{if } x \in Y, \text{ then } x \in X.$$

We must show that

$$\text{if } w \in Y, \text{ then } w \in X.$$

Suppose $w \in Y$. We must show that $w \in X$. There are three cases to consider.

- Suppose $w = \%$. Then $w = \% \in X$, by part (1) of the definition of X.
- Suppose $w = \mathtt{0}t$, for some $t \in \{\mathtt{0}, \mathtt{1}\}^*$. Since $1 + \textbf{diff}\, t = \textbf{diff}(\mathtt{0}t) = \textbf{diff}\, w = 0$, we have that $\textbf{diff}\, t = -1$. Let $u \in \{\mathtt{0}, \mathtt{1}\}^*$ be the shortest prefix of t such that $\textbf{diff}\, u \leq -1$, and let $v \in \{\mathtt{0}, \mathtt{1}\}^*$ be such that $t = uv$. Then $u \neq \%$, so that $u = xb$ for some $x \in \{\mathtt{0}, \mathtt{1}\}^*$ and $b \in \{\mathtt{0}, \mathtt{1}\}$. Hence $t = uv = xbv$. Since x is a shorter prefix of t than u, we have that $\textbf{diff}\, x \geq 0$.

  Suppose, toward a contradiction, that $b = \mathtt{0}$. Since $\textbf{diff}\, x + 1 = \textbf{diff}(x\mathtt{0}) = \textbf{diff}(xb) = \textbf{diff}\, u \leq -1$, we have that $\textbf{diff}\, x \leq -2$. But $\textbf{diff}\, x \geq 0$—contradiction. Thus $b = \mathtt{1}$.

  Summarizing, we have that $u = xb = x\mathtt{1}$, $t = uv = x\mathtt{1}v$, $w = \mathtt{0}t = \mathtt{0}x\mathtt{1}v$, $\textbf{diff}\, t = -1$ and $\textbf{diff}\, u \leq -1$. Since $\textbf{diff}\, x + -2 = \textbf{diff}(x\mathtt{1}) = \textbf{diff}\, u \leq -1$, we have that $\textbf{diff}\, x \leq 1$. But $\textbf{diff}\, x \geq 0$, and thus we have that $\textbf{diff}\, x \in \{0, 1\}$. Hence there are two sub-cases to consider.

  - Suppose $\textbf{diff}\, x = 0$. Because $-2 + \textbf{diff}\, v = 0 + -2 + \textbf{diff}\, v = \textbf{diff}(x\mathtt{1}v) = \textbf{diff}\, t = -1$, we have that $\textbf{diff}\, v = 1$. Since $\textbf{diff}\, v \geq 1$, Lemma PS2.1.1 tells us that $v = y\mathtt{0}z$, for some $y, z \in \{\mathtt{0}, \mathtt{1}\}^*$ such that $\textbf{diff}\, y = 0$ and $\textbf{diff}\, z = \textbf{diff}\, v - 1$. Hence $w = \mathtt{0}x\mathtt{1}v = \mathtt{0}x\mathtt{1}y\mathtt{0}z$ and $\textbf{diff}\, z = 0$. Since $\textbf{diff}\, x = \textbf{diff}\, y = \textbf{diff}\, z = 0$, we have that $x, y, z \in Y$. Because x, y and z are proper substrings of w, the inductive hypothesis tells us that $x, y, z \in X$. By part (3) of the definition of X, we have that $\mathtt{0}x\mathtt{1}y\mathtt{0} \in X$. Thus, by part (5) of the definition of X, we can conclude that $w = \mathtt{0}x\mathtt{1}y\mathtt{0}z = (\mathtt{0}x\mathtt{1}y\mathtt{0})z \in X$.
  - Suppose $\textbf{diff}\, x = 1$. Since $\textbf{diff}\, x \geq 1$, Lemma PS2.1.1 tells us that $x = s\mathtt{0}y$, for some $s, y \in \{\mathtt{0}, \mathtt{1}\}^*$ such that $\textbf{diff}\, s = 0$ and $\textbf{diff}\, y = \textbf{diff}\, x - 1$. Thus $\textbf{diff}\, y = 1 - 1 = 0$ and $w = \mathtt{0}x\mathtt{1}v = \mathtt{0}s\mathtt{0}y\mathtt{1}v$. We have that $\textbf{diff} v = 1 + 0 + 1 + 0 + -2 + \textbf{diff} v = \textbf{diff}(\mathtt{0}s\mathtt{0}y\mathtt{1}v) = \textbf{diff} w = 0$. Since $\textbf{diff}\, s = \textbf{diff}\, y = \textbf{diff}\, v = 0$, we have that $s, y, v \in Y$. Thus, because s, y and v are proper substrings of w, the inductive hypothesis tells us that $s, y, v \in X$. By part (2), of the definition of X, we have $\mathtt{0}s\mathtt{0}y\mathtt{1} \in X$. Thus by part (5) of the definition of X, we can conclude that $w = \mathtt{0}s\mathtt{0}y\mathtt{1}v = (\mathtt{0}s\mathtt{0}y\mathtt{1})v \in X$.

- Suppose $w = 1t$, for some $t \in \{0,1\}^*$. Since $-2 + \textbf{diff}\, t = \textbf{diff}(1t) = \textbf{diff}\, w = 0$, we have that $\textbf{diff}\, t = 2$. Because $\textbf{diff}\, t \geq 1$, Lemma PS2.1.1 tells us that $t = x0u$, for some $x, u \in \{0,1\}^*$ such that $\textbf{diff}\, x = 0$ and $\textbf{diff}\, u = \textbf{diff}\, t - 1$. Hence $\textbf{diff}\, u = 1$. Because $\textbf{diff}\, u \geq 1$, Lemma PS2.1.1 tells us that $u = y0z$, for some $y, z \in \{0,1\}^*$ such that $\textbf{diff}\, y = 0$ and $\textbf{diff}\, z = \textbf{diff}\, u - 1$. Hence $\textbf{diff}\, z = 0$.

  Summarizing, we have that $w = 1t = 1x0u = 1x0y0z$ and $x, y, z \in Y$. Since x, y and z are proper substrings of w, the inductive hypothesis tells us that $x, y, z \in X$. By part (4) of the definition of X, we have that $1x0y0 \in X$. Thus, by part (5) of the definition of X, we can conclude that $w = 1x0y0z = (1x0y0)z \in X$.

Problem 2

See the course website for the file `ps2-explain.sml`. Here is how `explain` was tested:

```
- use "ps2-framework.sml";
[opening ps2-framework.sml]
exception Error
val zero = - : sym
val one = - : sym
val isZero = fn : sym -> bool
val isOne = fn : sym -> bool
val diffSym = fn : sym -> int
val diff = fn : str -> int
val validStr = fn : str -> bool
datatype expl
  = Rule1
  | Rule2 of expl * expl
  | Rule3 of expl * expl
  | Rule4 of expl * expl
  | Rule5 of expl * expl
val strExplained = fn : expl -> str
val printExplanation = fn : expl -> unit
val test = fn : (str -> expl) -> str -> unit
val it = () : unit
- use "ps2-explain.sml";
[opening ps2-explain.sml]
val shortest = fn : (int -> bool) -> str -> str * str
val shortestPositive = fn : str -> str * str
val shortestNegative = fn : str -> str * str
val splitPositive = fn : str -> str * str
val explain = fn : str -> expl
val it = () : unit
- val doit = test explain;
val doit = fn : str -> unit
- doit(Str.fromString "%");
% is in X, by rule (1)
val it = () : unit
```

```
- doit(Str.fromString "001");
001 = 001 @ % is in X, by rule (5)
  001 = 0 @ % @ 0 @ % @ 1 is in X, by rule (2)
    % is in X, by rule (1)
    % is in X, by rule (1)
  % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "010");
010 = 010 @ % is in X, by rule (5)
  010 = 0 @ % @ 1 @ % @ 0 is in X, by rule (3)
    % is in X, by rule (1)
    % is in X, by rule (1)
  % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "100");
100 = 100 @ % is in X, by rule (5)
  100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
    % is in X, by rule (1)
    % is in X, by rule (1)
  % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "001010");
001010 = 001 @ 010 is in X, by rule (5)
  001 = 0 @ % @ 0 @ % @ 1 is in X, by rule (2)
    % is in X, by rule (1)
    % is in X, by rule (1)
  010 = 010 @ % is in X, by rule (5)
    010 = 0 @ % @ 1 @ % @ 0 is in X, by rule (3)
      % is in X, by rule (1)
      % is in X, by rule (1)
    % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "010100");
010100 = 010 @ 100 is in X, by rule (5)
  010 = 0 @ % @ 1 @ % @ 0 is in X, by rule (3)
    % is in X, by rule (1)
    % is in X, by rule (1)
  100 = 100 @ % is in X, by rule (5)
    100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
      % is in X, by rule (1)
      % is in X, by rule (1)
    % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "100001");
100001 = 100 @ 001 is in X, by rule (5)
  100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
    % is in X, by rule (1)
    % is in X, by rule (1)
```

```
  001 = 001 @ % is in X, by rule (5)
    001 = 0 @ % @ 0 @ % @ 1 is in X, by rule (2)
      % is in X, by rule (1)
      % is in X, by rule (1)
    % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "100000011010");
100000011010 = 100 @ 000011010 is in X, by rule (5)
  100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
    % is in X, by rule (1)
    % is in X, by rule (1)
  000011010 = 000011 @ 010 is in X, by rule (5)
    000011 = 0 @ % @ 0 @ 001 @ 1 is in X, by rule (2)
      % is in X, by rule (1)
      001 = 001 @ % is in X, by rule (5)
        001 = 0 @ % @ 0 @ % @ 1 is in X, by rule (2)
          % is in X, by rule (1)
          % is in X, by rule (1)
        % is in X, by rule (1)
    010 = 010 @ % is in X, by rule (5)
      010 = 0 @ % @ 1 @ % @ 0 is in X, by rule (3)
        % is in X, by rule (1)
        % is in X, by rule (1)
      % is in X, by rule (1)
val it = () : unit
- doit(Str.fromString "110001000001000111000");
110001000001000111000 = 110001000 @ 001000111000 is in X, by rule (5)
  110001000 = 1 @ 100 @ 0 @ 100 @ 0 is in X, by rule (4)
    100 = 100 @ % is in X, by rule (5)
      100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
        % is in X, by rule (1)
        % is in X, by rule (1)
      % is in X, by rule (1)
    100 = 100 @ % is in X, by rule (5)
      100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
        % is in X, by rule (1)
        % is in X, by rule (1)
      % is in X, by rule (1)
  001000111000 = 001 @ 000111000 is in X, by rule (5)
    001 = 0 @ % @ 0 @ % @ 1 is in X, by rule (2)
      % is in X, by rule (1)
      % is in X, by rule (1)
    000111000 = 000111000 @ % is in X, by rule (5)
      000111000 = 0 @ 001 @ 1 @ 100 @ 0 is in X, by rule (3)
        001 = 001 @ % is in X, by rule (5)
          001 = 0 @ % @ 0 @ % @ 1 is in X, by rule (2)
            % is in X, by rule (1)
            % is in X, by rule (1)
```

```
          % is in X, by rule (1)
        100 = 100 @ % is in X, by rule (5)
          100 = 1 @ % @ 0 @ % @ 0 is in X, by rule (4)
            % is in X, by rule (1)
            % is in X, by rule (1)
          % is in X, by rule (1)
      % is in X, by rule (1)
val it = () : unit
```

Note that the last two tests produce explanations using all five rules of X's definition.