Assignment 1

System Modelling and Design

Charlie Bradford z5114682

March 28, 2019

1 Specification

Define the predicate $P(l, a, b, m)$ to mean:

$$\forall x \in 1..(l-1); a[x] \neq a[x-1] \Rightarrow \exists j \in 1..m.(b[j] = a[x] \wedge b[j-1] = a[x-1] \wedge P(x, a, b, j-1))$$

P Checks recursively that all transitions between one string and another is preserved in order (by it's recursive nature). As every element in a either come before a different element or after a different element (or both), P ensures that all elements in a up to $a[i]$ are also in b.
Define the predicate $Q(l, b)$ to mean:

$$\forall x \in 1..l(b[x] \neq b[x-1])$$

This is the uniqueness predicate and ensures that there are no repeats in b.
Define the predicate $S(t, v, r)$ to mean:

$$\exists! x; t[x] = 0 \Rightarrow \forall m \in 0..(min(x, r) - 1)(t[m] = v[m])$$

This is the string equality predicate for use in loops.

Precondition: $\{n = |a| \wedge n \geq 0\}$

Postcondition: $\{k = |b| \wedge P(n, a, b, k) \wedge Q(k, b)\}$

An assignment axiom for 2-dimensional array must also be stated. In the following proofs $\{\psi[{}^{a\,:\,x\,\mapsto\,y\,\mapsto\,z}/_{a}]\}a[x][y] := z\{\psi\}$ is used.

$$\forall x \in \mathbb{R}|a[x] \in a(\forall y \in \mathbb{R}|a[x][y] \in a[x](\{\psi[{}^{a\,:\,x\,\mapsto\,y\,\mapsto\,z}/_{a}]\}a[x][y] := z\{\psi\}))$$

2 Implementation

```
                                                  $\{n = |a| \land n \geq 0\} \Rightarrow I[{}^{0}/_{1}]$
$i := 0;$
**while** $a[0][i] \neq 0$ **do**                  $I \land a[0][i] \neq 0 \Rightarrow I[{}^{i+1}/_{i}][{}^{b\,:\,0 \mapsto i \mapsto a[0][i]}/_{b}]$
  $b[0][i] := a[0][i];$                           $I[{}^{i+1}/_{i}]$
  $i := i + 1;$                                   $I$
 **od**                                           $I \land a[0][i] = 0 \Rightarrow J[{}^{0}/_{k}][{}^{0}/_{i}]$
$i := 0;$                                         $\{J[{}^{0}/_{k}]\}$
$k := 0;$                                         $\{J\}$
**while** $i < n$ **do**                          $\{J \land i < n \land \neg S(a[i], b[k], \infty)\}$
                                                   $\Rightarrow J[{}^{k+1}/_{k}][{}^{b\,:\,k \mapsto a[i]}/_{b}][{}^{i+1}/_{i}]$
                                                  $\{J \land i < n \land S(a[i], b[k], \infty)\} \Rightarrow J[{}^{i+1}/_{i}]$
  $copy := False;$                                $J[{}^{i+1}/_{i}] \land i < n \Rightarrow K[{}^{0}/_{j}][{}^{False}/_{copy}]$
  $j := 0;$                                       $K$
  **while** $a[i][j] \neq 0 \land b[k][j] \neq 0 \land \neg copy$ **do**   $K \land \neg copy \land a[i][j] = b[k][j]$
                                                   $\Rightarrow K[{}^{j+1}/_{j}]$
                                                  $K \land \neg copy \land b[i][j] \neq b[k][j] \Rightarrow K[{}^{j+1}/_{j}][{}^{True}/_{copy}]$
                                                  $K \land \neg copy \land b[i][j] = 0 \land b[k][j] = 0 \Rightarrow K[{}^{j+1}/_{j}] \land \neg copy$
    **if** $a[i][j] \neq b[k][j]$ **then**
      $copy := True;$
    **fi**
    $j := j + 1;$                                 $K$
  **od**
  **if** $copy$ **then**                          $K[{}^{True}/_{copy}] \Rightarrow L[{}^{k+1}/_{k}][{}^{0}/_{j}]$
    $j := 0;$                                     $L[{}^{k+1}/_{k}]$
    $k := k + 1;$                                 $L$
    **while** $a[i][j] \neq 0$ **do**             $L \land a[i][j] \neq 0 \Rightarrow L[{}^{j+1}/_{j}][{}^{b\,:\,k \mapsto j \mapsto a[i][j]}/_{b}]$
      $b[k][j] := a[i][j];$                       $L[{}^{j+1}/_{j}]$
      $j := j + 1;$                               $L$
    **od**
  **fi**
  $i := i + 1;$                                   $J$
**od**                                            $J \land i > n \Rightarrow \{k = |b| \land P(n, a, b, k) \land Q(k, b)\}[{}^{k+1}/_{k}]$
$k := k + 1$
```

2.1 Loop Invariants

$I \ = \{S(a[0], b[0], i)\}$

$J \ = \{0 \leq i \leq n \land k + 1 = |b| \land P(i, a, b, k) \land Q(k, b)\}$

$K \ = \{copy \oplus S(a[i], b[k], j)\}(\oplus =$ logical xor)

$L \ = \{S(a[i], b[k], j)\}$

2.2 Implication 1

$$\{n = |a| \land n \geq 1\} \Rightarrow I[{}^{0}/_{i}]$$

We begin by unpacking I.

$$\begin{aligned}
&n = |a| \land n \geq 1\\
\Rightarrow &S(a[0], b[0], i)[{}^{0}/_{i}]\\
\Leftrightarrow &S(a[0], b[0], 0)\\
\Leftrightarrow &\exists! x; a[0][x] = 0 \Rightarrow \forall m \in 0..(min(x, 0) - 1)(a[0][m] = b[0][m])\\
\Leftrightarrow &\exists! x; a[0][x] = 0 \Rightarrow \forall m \in 0..(-1)(a[0][m] = b[0][m])
\end{aligned}$$

The RHS is true as there are no values which to compare. Thus the implication is true.

2.3 Implication 2

$$I \land a[0][i] \neq 0 \Rightarrow I[{}^{i+1}/_{i}][{}^{b\,:\,0 \mapsto i \mapsto a[0][i]}/]$$

We begin by unpacking the invariant and performing substitutions.

$$\begin{aligned}
&S(a[0], b[0], i) \land a[0][i] \neq 0\\
\Rightarrow &S(a[0], b[0], i)[{}^{i+1}/_{i}][{}^{b\,:\,0 \mapsto i \mapsto a[0][i]}/_{b}]
\end{aligned}$$

2.4 Implication 3

$$I \land a[0][i] = 0 \Rightarrow J[{}^{1}/_{i}][{}^{0}/_{k}]$$

We begin by unpacking the invariants and performing any substitutions.

$$n = |a| \land n \geq 1 \land \forall l \in 0..(i-1)(a[0][l] = b[0][l]) \land a[0][i] = 0$$

On the RHS, the first cojunct is trivial, and the last is vacuous. Given that i is the last character in $a[0]$ and that everything before i is stored in $b[0]$ we know that the size of b is one (as it was empty before), thus we know that the second last conjuct is also true. As far as the predicate is concerned, there is no pair of different numbers from 1 to 1 so it returns true regardless.

2.5 Implication 4

$$J \land i < n \land \neg S(a[i], b[k], \infty) \Rightarrow J[{}^{k+1}/_{k}][{}^{b\,:\,k \mapsto a[i]}/_{b}][{}^{i+1}/_{i}]$$

This considers the case that a newline has been found and must be added to b.

$$\begin{aligned}
&0 \leq i \leq n \land k+1 = |b| \land P(i, a, b, k) \land Q(k, b) \land i < n \land \neg S(a[i], b[k], \infty)\\
\Rightarrow &\{0 \leq i \leq n \land k+1 = |b| \land P(i, a, b, k) \land Q(k, b)\}[{}^{k+1}/_{k}][{}^{b\,:\,k \mapsto a[i]}/_{b}][{}^{i+1}/_{i}]\\
\Leftrightarrow &\leq i+1 \leq n \land k+2 = |(b : k+1 \mapsto a[i])| \land P(i+1, a, (b : k+1 \mapsto a[i]), k+1) \land Q(k+q, (b : k+1 \mapsto a[i]))
\end{aligned}$$

We know that $i < n$ from the LHS and so can discharge that and $0 \leq i \leq n$ from the LHS and $0 \leq i+1 \leq n$ from the RHS. We also know that, having performed an assignment to values within $b[k+1]$, $|b| = k+2$. This leaves us with the predicates P and Q.

$$\begin{aligned}
&P(i, a, b, k) \land Q(k, b) \land \neg S(a[i], b[k], \infty)\\
\Rightarrow &P(i+1, a, (b : k+1 \mapsto a[i]), k+1) \land Q((b : k+1 \mapsto a[i]), k+1)
\end{aligned}$$

First let us focus on P. On the RHS P considers cases for $x \in 1..i$ and $m \in 1..(k+1)$, however our we already have cases $x \in 1..(i-1)$ and $j \in 1..k$ present on our LHS. Thus we need only prove $P(i+1, a, (b : k+1 \mapsto a[i]), k+1)$ for cases where $x = i$ and $j = k+1$.

$$\begin{aligned}
&\forall x \in 1..(i-1); a[x] \neq a[x-1] \Rightarrow \exists j \in 1..k.(b[j] = a[x] \land b[j-1] = a[x-1] \land P(x, a, b, j-1))\\
\Rightarrow &\forall x \in 1..i; a[x] \neq a[x-1] \Rightarrow \exists j \in 1..(k+1).((b : k+1 \mapsto a[i])[j] = a[x] \land (b : k+1 \mapsto a[i])[j-1] = a[x-1] \land P(x, a, (b : k+1\\
\Leftrightarrow &a[i] \neq a[i-1] \Rightarrow (b : k+1 \mapsto a[i])[k+1] = a[i] \land (b : k+1 \mapsto a[i])[k] = a[i-1] \land P(i, a, (b : k+1 \mapsto a[i]), k)
\end{aligned}$$

The last conjuct is considered in the RHS so can be discarded. And, given $P(i, a, b, k)$, we know that the element that last occurs in $a[i-1]$ must also occur in $b[k]$. This leaves $(b : k+1 \mapsto a[i])[k+1] = a[i]$ which is tautological.
Now considering Q. Likewise with P, the values of $x \in 1..k$ have be considered, so we need only prove Q for $x = k+1$.

$$\begin{aligned}
&\forall x \in 1..k(b[x] \neq b[x-1]) \wedge \neg S(a[i], b[k], \infty)\\
\Rightarrow&\forall x \in 1..(k+1)((b : k+1 \mapsto a[i])[x] \neq (b : k+1 \mapsto a[i])[x-1])\\
\Leftrightarrow&(b : k+1 \mapsto a[i])[k+1] \neq (b : k+1 \mapsto a[i])[k])\\
\Leftrightarrow&a[i] \neq b[k]
\end{aligned}$$

Which was shown on the LHS.

2.6 Implication 5

$$J \wedge i < n \wedge S(a[i], b[k], \infty) \Rightarrow J[^{i+1}/_i]$$

This considers the case that a line is repeated, and thus, ignored.

$$\begin{aligned}
&0 \leq i \leq n \wedge k+1 = |b| \wedge P(i, a, b, k) \wedge Q(k, b) \wedge i < n \wedge S(a[i], b[k], \infty)\\
\Rightarrow&\{0 \leq i \leq n \wedge k+1 = |b| \wedge P(i, a, b, k) \wedge Q(k, b)\}[^{i+1}/_i]\\
\Leftrightarrow&0 \leq i+1 \leq n \wedge k+1 = |b| \wedge P(i+1, a, b, k) \wedge Q(k, b)
\end{aligned}$$

As in implication four we can discard the conjucts $0 \leq i \leq n$ and $i < n$ from the LHS, and $0 \leq i+1 \leq n$ from the RHS. k and b both remain unchanged so we can discharge both conjucts that only concern those variables from both the LHS and RHS. This leaves only P and S.

$$\begin{aligned}
&P(i, a, b, k) \wedge S(a[i], b[k], \infty)\\
\Rightarrow&P(i+1, a, b, k)
\end{aligned}$$

The only difference between the LHS and RHS would be if $a[i] \neq a[i-1]$, however, given $P(i, a, b, k)$ we know that $b[k] = a[i-1]$, and, given $S(a[i], b[k], \infty)$, we know that $b[k] = a[i]$, so the LHS and RHS are logically equivalent.

2.7 Implication 6

$$J[^{i+1}/_i] \wedge i < n \Rightarrow K[^0/_j][^{False}/_{copy}]$$

$$\begin{aligned}
&J[^{i+1}/_i] \wedge i < n\\
\Leftrightarrow&0 \leq i+1 \leq n \wedge k+1 = |b| \wedge P(i+1, a, b, k) \wedge Q(k, b)\\
\Rightarrow&K[^0/_j][^{False}/_{copy}]\\
\Leftrightarrow&False \oplus S(a[i], b[k], 0)
\end{aligned}$$

The last conjuct of the RHS is vacuously true for $j = 0$. $(False \oplus True) = True$, so the the RHS is always true, regardless of the LHS. Thus the implication is always true.

2.8 Implication 7

$$K \wedge \neg copy \wedge a[i][j] = b[k][j] \Rightarrow K[^{j+1}/_j]$$

$$\begin{aligned}
&K \wedge \neg copy \wedge a[i][j] = b[k][j]\\
\Leftrightarrow&(copy \oplus S(a[i], b[k], j)) \wedge \neg copy \wedge a[i][j] = b[k][j]\\
\Leftrightarrow&S(a[i], b[k], j) \wedge a[i][j] = b[k][j]\\
\Rightarrow&K[^{j+1}/_j]\\
\Leftrightarrow&copy \oplus S(a[i], b[[k], j+1)
\end{aligned}$$

Expanding the RHS.

$$\begin{aligned}
&S(a[i], b[k], j) \wedge a[i][j] = b[k][j]\\
\Rightarrow\ &\exists! x; t[x] = 0 \Rightarrow m \in 0..j(a[i][m] = b[k][m])\\
\Leftrightarrow\ &S(a[i], b[k], j) \wedge a[i][j] = b[k][j]
\end{aligned}$$

The LHS is equivalent to the RHS.

2.9 Implication 8

$$K \wedge \neg copy \wedge a[i][j] \neq b[k][j] \Rightarrow K[{}^{j+1}\!/_{j}][{}^{True}\!/_{copy}]$$

$$\begin{aligned}
&(copy \oplus S(a[i], b[k], j)) \wedge \neg copy \wedge a[i][j] \neq b[k][j] \\
\Leftrightarrow &S(a[i], b[k], j) \wedge a[i][j] \neq b[k][j] \\
\Rightarrow &True \oplus S(a[i], b[k], j+1) \\
\Leftrightarrow &\neg S(a[i], b[k], j+1)
\end{aligned}$$

Exactly as with above S for all cases up to j is considered in the LHS, so we can discard S from the LHS and simultaneously unpack and discard what we already know from S on the RHS.

$$\begin{aligned}
&a[i][j] \neq b[k][j] \\
\Rightarrow &\forall m \in j..j(a[i][m] \neq b[k][m]) \\
\Leftrightarrow &a[i][j] \neq b[k][j]
\end{aligned}$$

The LHS is equivalent to the RHS.

2.10 Implication 9

$$K \wedge \neg copy \wedge a[i][j] = 0 \wedge b[k][j] = 0 \Rightarrow K[{}^{j+1}\!/_{j}] \wedge \neg copy$$

$$\begin{aligned}
&(copy \oplus S(a[i], b[k], j)) \wedge \neq copy \wedge a[i][j] = 0 \wedge b[k][j] = 0 \\
\Leftrightarrow &S(a[i], b[k], j) \wedge a[i][j] = 0 \wedge b[k][j] = 0 \\
\Rightarrow &(copy \oplus S(a[i], b[k], j+1)) \wedge \neg copy \\
\Leftrightarrow &S(a[i], b[k], j+1)
\end{aligned}$$

The predicate S finds a value for x where the $t[x] = 0$, i.e. the end of the null terminated string. It then tests equality of each character in the string up until one less than the minimum of x and a supplied r. However in this case $r = x$, and $r = j + 1$, and therefore $j + 1 < x$. So now we know that the predicate is true for values up to $j - 1$ and we need only prove it for j.

$$\begin{aligned}
&S(a[i], b[k], j) \wedge a[i][j] = 0 \wedge b[k][j] = 0 \\
\Leftrightarrow &\forall m \in 0..(j-1)(a[m] = b[m]) \wedge a[i][j] = 0 \wedge b[k][j] = 0 \\
\Rightarrow &\forall m \in 0..j(a[m] = b[m])
\end{aligned}$$

Discard the redundant.

$$\begin{aligned}
&a[i][j] = 0 \wedge b[k][j] = 0 \\
\Rightarrow &a[i][j] = b[k][j]
\end{aligned}$$

The values of $a[i][j]$ and $b[k][j]$ are known to be the same from the LHS so the statement is true.

2.11 Implication 10

$$K[{}^{True}\!/_{copy}] \Rightarrow L[{}^{k+1}\!/_{k}][{}^{0}\!/_{j}]$$

$$\begin{aligned}
&True \oplus S(a[i], b[k], j) \\
\Leftrightarrow &\neg S(a[i], b[k], j) \\
\Rightarrow &S(a[i], b[k+1], 0)
\end{aligned}$$

As shown previously, S is vacuously true for third argument 0. (LHS$\Rightarrow True) = True$

2.12 Implication 11

$$L \wedge a[i][j] \neq 0 \Rightarrow L[{}^{j+1}\!/_{j}][b : k \mapsto j \mapsto a[i][j]b]$$

An equivalent statement was proved in Implication 2, see section 2.3 for more details.

2.13 Implication 12

$$J \land i \geq n \Rightarrow \text{Postcondition}[{}^{k+1}/_{k}]$$

Unpack

$$\begin{aligned}&0 \leq i \leq n \land k+1 = |b| \land P(i,a,b,k) \land Q(k,b) \land i \geq n\\ \Rightarrow &k+1 = |b| \land P(n,a,b,k) \land Q(k,b)\end{aligned}$$

Things look pretty clear. $0 \leq i \leq n \land i \geq n$ can be resolved to simply $i = n$.

$$\begin{aligned}&i = n \land k+1 = |b| \land P(i,a,b,k) \land Q(k,b)\\ \Rightarrow &k+1 = |b| \land P(n,a,b,k) \land Q(k,b)\end{aligned}$$

Now the only difference is the arguments passed to P. But we said they are the same. So the statements are logically equivalent and thus LHS ⇒ RHS.

3 Program

```
#include <string.h>

typedef int bool;
#define TRUE 1
#define FALSE 0 /* I know fake booleans are gross but */
                /* I'm  trying to stick to the script */

int uniq(unsigned int n, char *a[], char *b[])
{
    strcpy(b[0], a[0]);
    int i = 0;
    int k = 0;
    while(i < n)
    {
        bool copy = FALSE;
        int j = 0;
        while (a[i][j] != '\0' && b[k][j] != '\0' && !copy)
        {
            if (a[i][j] != b[k][j])
            {
                copy = TRUE;
            }
            j = j + 1;
        }
        if (copy)
        {
            k = k + 1;
            strcpy(b[k], a[i]);
        }
        i = i + 1;
    }
    return k + 1;
}
```

Changes made:

- Assigning a value directly to a char in a string is undefined in C, so I had to use the strcpy() function contained in string.h for the loops with invariants I and L.