Suggested Solutions to HW #2

1 Consider the following recurrence relation:

$$\begin{cases} T(0) &= 0 \\ T(1) &= 1 \\ T(h) &= T(h-1)+T(h-2)+1, \quad h \geq 2 \end{cases}$$

Prove by induction the relation $T(h) = F(h+2) - 1$ where $F(n)$ is the n-th Fibonacci number ($F(1) = 1$, $F(2) = 1$, and $F(n) = F(n-1) + F(n-2)$, for $n \geq 3$).

Solution. (Ming-Hsien Tsai)

Base case: When $h = 0$, $T(0) = F(2) - 1$ because $T(0) = 0$ and $F(2) = 1$. When $h = 1$, $T(1) = F(3) - 1$ because $T(1) = 1$ and $F(3) = F(2) + F(1) = 2$.

Inductive step: Assume $T(h) = F(h+2) - 1$ and $T(h+1) = F(h+3) - 1$ for some $h \geq 0$. Then

$$\begin{aligned} T(h+2) &= T(h+1) + T(h) + 1 && \text{(Definition of T)} \\ &= (F(h+3)-1) + (F(h+2)-1) + 1 && \text{(Inductive hypothesis)} \\ &= F(h+4) - 1 && \text{(Definition of F)} \end{aligned}$$

□

4 (2.39) Design an algorithm to convert a binary number to a decimal number. The algorithm should be the opposite of algorithm *Convert_to_Binary* (see Fig. 1). The input is an array of bits b of length k, and the output is a number n. Prove the correctness of your algorithm by using a loop invariant.

Algorithm Convert_to_Binary(n):
Input: n (a positive integer).
Output: b (an array of bits corresponding to the binary representation of n).

```
begin
    t := n ;
    k := 0 ;
    while t > 0 do
        k := k + 1 ;
        b[k] := t mod 2 ;
        t := t div 2 ;
end
```

Figure 1: Algorithm *Convert_to_Binary*.

Solution. (Modified by Yi-Wen Chang)

Algorithm Convert_to_Decimal (b, k);
Input: b (an array of bits containing a binary number), k (the array size of b)
Output: dec (a positive integer corresponding to the decimal number of array b)
begin

$dec := 0$;

$i := k$;

while $i > 0$ **do**

$dec \ := \ dec \times 2 + b[i]$;

$i \ := \ i - 1$;

end

Let $Inv(n, dec, k, b)$ denote the following assertion:

$n = dec \times 2^i + m$ and $i \geq 0$,

where n is the number represented by b, i.e.,

$$n = \begin{cases} 0 & \text{if } k = 0 \\ b[k] \times 2^{k-1} + b[k-1] \times 2^{k-2} + \cdots + b[1] \times 2^0 & \text{if } k \geq 1 \end{cases}$$

and m is the number represented by b from position i to 0, i.e.,

$$m = \begin{cases} 0 & \text{if } i = 0 \\ b[i] \times 2^{i-1} + b[i-1] \times 2^{i-2} + \cdots + b[1] \times 2^0 & \text{if } i \geq 1 \end{cases}$$

Claim: $Inv(n, dec, i, b)$ is a loop invariant of the while loop, assuming that n is non-negative. (The invariant is sufficient to deduce that, when the program terminates, $i = 0$ and so dec stores the decimal n represented by b.)

Proof: The proof is by induction on the number of times the loop is executed. More specifically, we show that (1) the assertion is true when the flow of control reaches the loop for the first time and (2) given that the assertion is true and the loop condition holds, the assertion will remain true after the next iteration (i.e., after the loop body is executed once more).

(1) When the flow of control reaches the loop for the first time, $dec = 0$ and $i = k$. With m denoting the binary number represented by b from position k to 0, $dec \times 2^i + m = 0 \times 2^k + m = 0 + n = n$ (when $i = k$, m equals n, trivially) and $i \geq 0$. Therefore, the assertion $Inv(n, dec, i, b)$ holds.

(2) Assume that $Inv(n, dec, i, b)$ is true at the start of the next iteration and the loop condition $(i > 0)$ holds. Let n', dec', i', and b' denote respectively the values of n, t, i, and b after the next iteration. We need to show that $Inv(n', t', i', b')$ also holds.

From the loop body, we deduce the following relationship:

$$\begin{aligned} & i' = i - 1 \\ & b'[j] = b[j] \text{ for all } j \leq k \\ & dec' = dec \times 2 + b[i] \\ & m' = m - b[i] \times 2^{i-1} \\ & n' = n \text{ (the value of } n \text{ never changes)} \end{aligned}$$

Thus, we have:

$$
\begin{aligned}
dec' \times 2^{i'} + m' &= (dec \times 2 + b[i]) \times 2^{i-1} + m' \\
&= dec \times 2 \times 2^{i-1} + b[i] \times 2^{i-1} + m' \\
&= dec \times 2^{i} + b[i] \times 2^{i-1} + m - b[i] \times 2^{i-1} \\
&= dec \times 2^{i} + m = n = n'
\end{aligned}
$$

In addition, since $i > 0$ (given that the loop condition holds), $i' = i - 1 \geq 0$.Therefore, $Inv(n', t', k', b')$ holds after the next iteration. $\square$