CPS122 - OBJECT-ORIENTED SOFTWARE DEVELOPMENT

An Example of a Correctness Proof for the playOneRound() method of the coffee can problem

Notation: Let w and b = the *current* number of white and black beans, respectively
Let w_0 and b_0 = the *initial* number of white and black beans, respectively
Let w_i and b_i = the number of beans in the can *when the method is called*, respectively.

Class Invariant: $w \geq 0$ and $b \geq 0$ and $w + b \geq 1$ and $w \% 2 == w_0 \% 2$

```
/** Play one round of the "coffee can" game.  Draw two beans and put
 *  one back, as described by the rules above.
 *
 *  Precondition: there is more than one bean in the can
 *
 *  Postcondition: the number of beans in the can is reduced by 1, in
 *                      accordance with the rules of the game
 *   @return string describing what took place
 */
public String playOneRound()
{
```

// $w \geq 0$ and $b \geq 0$ and $w \% 2 == w_0 \% 2$ (from invariant)
// $w + b > 1$ (from precondition - stronger than invariant)
// $w_i = w$ and $b_i = b$ (definition of terms)

```
     String first = chooseBean();
```

// (All of the above) and
// ((first == “White” and $w > 0$) or (first == “Black” and $b > 0$))

```
     if (first == "White")
          whiteBeans --;
     else
          blackBeans --;
```

// $w \geq 0$ and $b \geq 0$ and $w + b \geq 1$ and $w + b = w_i + b_i - 1$ and
// ((first == “White” and $(w+1) \% 2 == w_0 \% 2$) or
// (first == “Black” and $w \% 2 == w_0 \% 2$))

```
     String second = chooseBean();
```

// (All the above) and
// ((second == “White” and $w > 0$) or (second == “Black” and $b > 0$))

```
     if (second == "White")
          whiteBeans --;
     else
          blackBeans --;
```

// $w \geq 0$ and $b \geq 0$ and $w + b \geq 0$ and $w + b = w_i + b_i - 2$ and
// ((first == second and $w \% 2 == w_0 \% 2$) or
// (first != second and $(w+1) \% 2 == w_0 \% 2$))

```
     String putBack;
     if (first == second)
     {
          putBack = "Black";
          blackBeans ++;
     }
     else
     {
          putBack = "White";
          whiteBeans ++;
     }
```

// w ≥ 0 and b ≥ 0 and w + b ≥ 1 and w % 2 == w_0 % 2 (invariant)

// w + b = w_i + b_i - 1 (postcondition)

// ∴ Invariant is preserved and postcondition is established

```
     return "Drew: " + first + ", " + second + ".  Put back: " + putBack;
}

/** Choose a single bean to draw
 *
 *   Preconditions:  there is at least one bean in the can
 *
 *   Postconditions: Return value is either "White" or "Black",
 *             and there is at least one bean in the can of that color
 *
 *   @return color of bean to draw
 */
private String chooseBean()
{
```

// w + b ≥ 1

```
     if (whiteBeans > 0 && blackBeans > 0)
          if (Math.random() < 0.5)
               return "White";
          else
               return "Black";
     else if (whiteBeans > 0)
          return "White";

     else // must be the case that blackBeans > 0
          return "Black";
```

// (result == “White” and w > 0) or (result = “Black” and b > 0)

```
}
```