CS5010 - Problem Set 07 - Test Results

pdp-group-elicaf-rosefox911

March 18, 2014

This test suite tests your implementation of Problem Set 07

1 File: pitchers.rkt

1.1 Test-Group: pitchers-to-list/list-to-pitchers (1 Points)

1/1

1.1.1 Test (predicate)

Input:

```
(pitchers-to-list (list-to-pitchers '((8 8) (0 5) (0 3))))
```

Output should match:

```
(listrep-equiv? '((8 8) (0 5) (0 3)))
```

Correct

1.2 Test-Group: pitchers-after-moves (5 Points)

5/5

Common Definitions

```
(define PITCHERS-0 (list-to-pitchers '((1 1))))


(define PITCHERS-1 (list-to-pitchers '((8 8) (0 5) (0 3))))


(define PITCHERS-2 (list-to-pitchers '((3 8) (5 5) (0 3))))


(define PITCHERS-3 (list-to-pitchers '((0 8) (3 3) (5 5))))


(define PITCHERS-4
(list-to-pitchers '((4 12) (5 5) (3 3) (0 2) (0 4) (0 24) (0 0))))
```

```
(define MOVES-1 (list (make-move 1 2)))

(define MOVES-2 (list (make-move 1 2) (make-move 2 1)))

(define MOVES-3 (list (make-move 1 3)))

(define MOVES-4 (list (make-move 2 5) (make-move 3 6)))

(define MOVES-5
(list
 (make-move 1 6)
 (make-move 2 6)
 (make-move 3 6)
 (make-move 6 7)
 (make-move 6 5)
 (make-move 5 3)
 (make-move 6 1)))
```

1.2.1 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-2 empty))
```

Output should match:

```
(listrep-equiv? '((3 8) (5 5) (0 3)))
```

Correct

1.2.2 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-1 MOVES-1))
```

Output should match:

```
(listrep-equiv? '((3 8) (5 5) (0 3)))
```

Correct

1.2.3 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-1 MOVES-2))
```

Output should match:

```
(listrep-equiv? '((8 8) (0 5) (0 3)))
```

Correct

1.2.4 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-3 MOVES-2))
```

Output should match:

```
(listrep-equiv? '((3 8) (0 3) (5 5)))
```

Correct

1.2.5 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-1 MOVES-3))
```

Output should match:

```
(listrep-equiv? '((5 8) (0 5) (3 3)))
```

Correct

1.2.6 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-3 MOVES-3))
```

Output should match:

```
(listrep-equiv? '((0 8) (5 5) (3 3)))
```

Correct

1.2.7 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-4 MOVES-4))
```

Output should match:

```
(listrep-equiv? '((4 12) (1 5) (0 3) (0 2) (4 4) (3 24) (0 0)))
```

Correct

1.2.8 Test (predicate, 1/2 partial points)

Input:

```
(pitchers-to-list (pitchers-after-moves PITCHERS-4 MOVES-5))
```

Output should match:

```
(listrep-equiv? '((8 12) (0 5) (3 3) (0 2) (1 4) (0 24) (0 0)))
```

Correct

1.3 Test-Group: solve (base cases and problem set examples) (3 Points)

3/3

1.3.1 Test (equality, 1/2 partial points)

Input:

```
(solve '(2) 1)
```

Expected Output:

```
false
```

Expected Output Value:

```
#f
```

Correct

1.3.2 Test (equality, 1/2 partial points)

Input:

```
(solve '(2) 2)
```

Expected Output:

```
empty
```

Expected Output Value:

```
()
```

Correct

1.3.3 Test (predicate, 1/2 partial points)

Input:

```
(solve '(8 5 3) 4)
```

Output should match:

```
(check-moves '(8 5 3) 4)
```

Correct

1.3.4 Test (predicate, 1/2 partial points)

Input:

```
(solve '(10 7 3) 5)
```

Output should match:

```
(check-moves '(10 7 3) 5)
```

Correct

1.4 Test-Group: solve (more) (5 Points)

5/5

1.4.1 Test (equality, 1/2 partial points)

Input:

```
(solve '(8 4 2) 3)
```

Expected Output:

```
false
```

Expected Output Value:

```
#f
```

Correct

1.4.2 Test (equality, 1 partial points)

Input:

```
(solve '(9 8 7) 4)
```

Expected Output:

```
false
```

Expected Output Value:

```
#f
```

Correct

1.4.3 Test (predicate, 1 partial points)

Input:

```
(solve '(8 5 3 0) 4)
```

Output should match:

```
(check-moves '(8 5 3 0) 4)
```

Correct

1.4.4 Test (predicate, 1 partial points)

Input:

```
(solve '(8 11/2 7/2) 9/2)
```

Output should match:

```
(check-moves '(8 11/2 7/2) 9/2)
```

Correct

1.4.5 Test (predicate, 1/2 partial points)

Input:

```
(solve '(8 5 3) 7)
```

Output should match:

```
(check-moves '(8 5 3) 7)
```

Correct

1.5 Test-Group: solve (long, but small) (1 Points)

1/1

1.5.1 Test (predicate)

Input:

```
(solve '(5 1 1 1 1) 2)
```

Output should match:

```
(check-moves '(5 1 1 1 1) 2)
```

Correct

2 Results

Successes: 19
Wrong Outputs: 0
Errors: 0
Achieved Points: 15
Total Points (rounded): 15/15