# Writing Test Cases II

Cai, Zelin Silvestre, Patrick

## 1 No Currency Test Cases - Original Test Cases

| Test Name | Input Vector | Expected Output |
|---|---|---|
| test dispense red | `<dispense_red()>` | "You need at least 5 cents to dispense a red gumball" |
| test dispense yellow | `<dispense_yellow()>` | "You need at least 10 cents to dispense a red gumball" |
| test return my change | `<return_my_change()>` | "There is no change to return" |

Table 1: Original "No Currency" Test Cases

Of these three test cases, no particular technique (e.g. control flow testing, data flow testing, etc.) was used in their creation.

# 2 No Currency Test Cases - Revised Test Cases

In general, the input domain associated with "no currency" testing involve invocations of any functions other than `insert()`.

## 2.1 Control Flow Testing

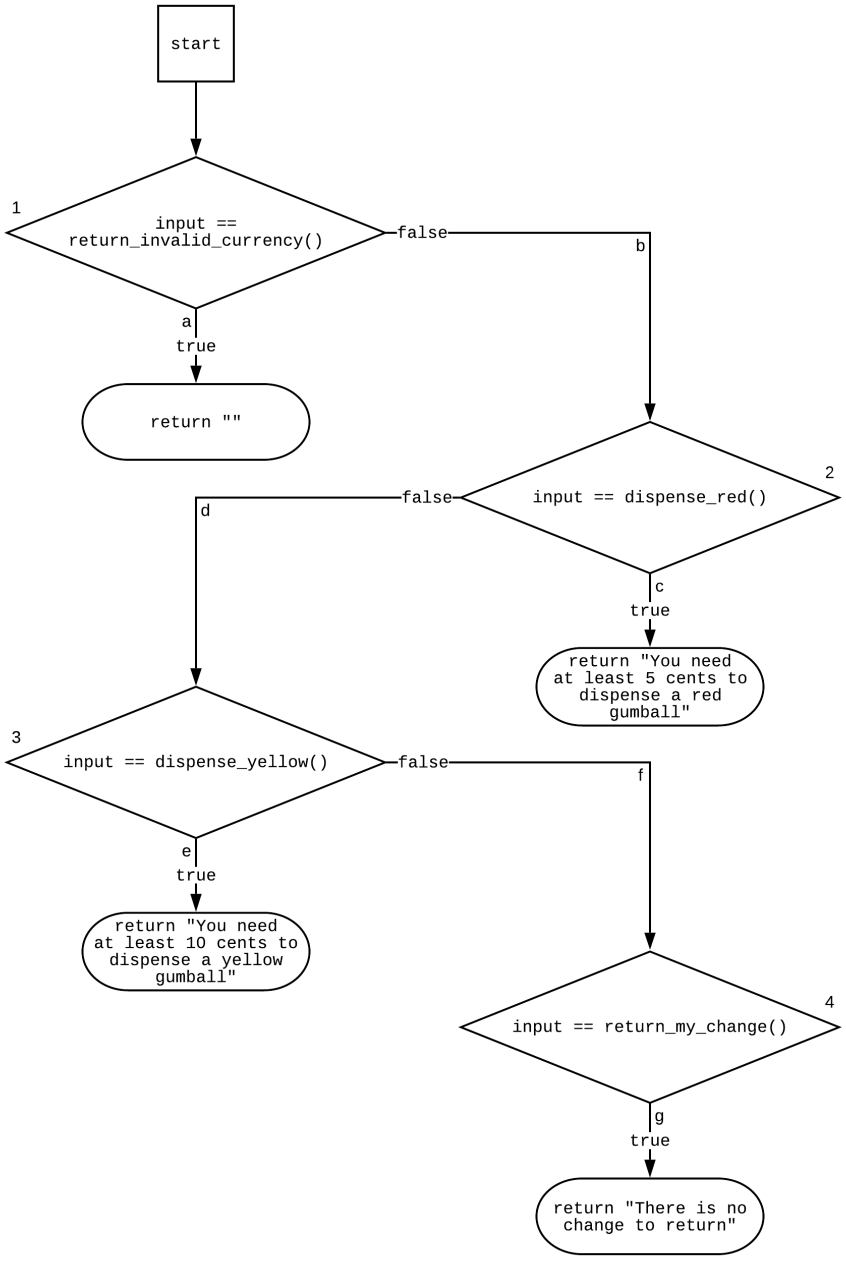


Figure 1: Gumball Machine Control Flow Diagram (No Currency Inserted)

Using the control flow diagram, we generate test cases to ensure statement coverage:

| Path | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|
| (a) | ✓ | | | | | | |
| (b, c) | | ✓ | ✓ | | | | |
| (b, d, e) | | ✓ | | ✓ | ✓ | | |
| (b, d, f, g) | | ✓ | | ✓ | | ✓ | ✓ |

Table 2: Control Flow Testing Statement Coverage

| Path | Input Vector | Expected Output |
|---|---|---|
| (a) | `<return_invalid_currency()>` | “” |
| (b, c) | `<dispense_red()>` | “You need at least 5 cents to dispense a red gumball” |
| (b, d, e) | `<dispense_yellow()>` | “You need at least 10 cents to dispense a yellow gumball” |
| (b, d, f, g) | `<return_my_change()>` | “There is no change to return” |

Table 3: Revised “No Currency” Test Cases - Control Flow Testing

## 2.2 Equivalence Class Testing

The input domain of “all functions other than `insert()`” can be partitioned based on the remaining functions:

- EC-01: `return_invalid_currency()`
- EC-02: `dispense_red()`
- EC-03: `dispense_yellow()`
- EC-04: `return_my_change()`

We generate a test case for each uncovered equivalence class:

| TC | Test Value | Expected Result | EC-01 | EC-02 | EC-03 | EC-04 |
|---|---|---|---|---|---|---|
| 01 | `<return_invalid_currency()>` | “” | ✓ | | | |
| 02 | `<dispense_red()>` | “You need at least 5 cents to dispense a red gumball” | | ✓ | | |
| 03 | `<dispense_yellow()>` | “You need at least 10 cents to dispense a yellow gumball” | | | ✓ | |
| 04 | `<return_my_change()>` | “There is no change to return” | | | | ✓ |

Table 4: Revised “No Currency” Test Cases - Equivalence Class Testing

# 3 Exact Currency Test Cases - Original Test Cases

| Test Name | Input Vector | Expected Output |
|---|---|---|
| test insert nickel dispense red | `<insert("nickel"), dispense_red()` | "Enjoy your red gumball" |
| test insert dime dispense yellow | `<insert("nickel"), dispense_yellow()>` | "Enjoy your yellow gumball" |
| test insert nickels dispense yellow | `<insert("nickel"), insert("nickel"), dispense_yellow()>` | "Enjoy your yellow gumball" |

Table 5: Original "Exact Currency" Test Cases

Of these three test cases, no particular technique (e.g. control flow testing, data flow testing, etc.) was used in their creation.

# 4 Exact Currency Test Cases - Revised Test Cases

In general, we assume the gumball machine has enough currency when applicable.

## 4.1 Conformance Testing

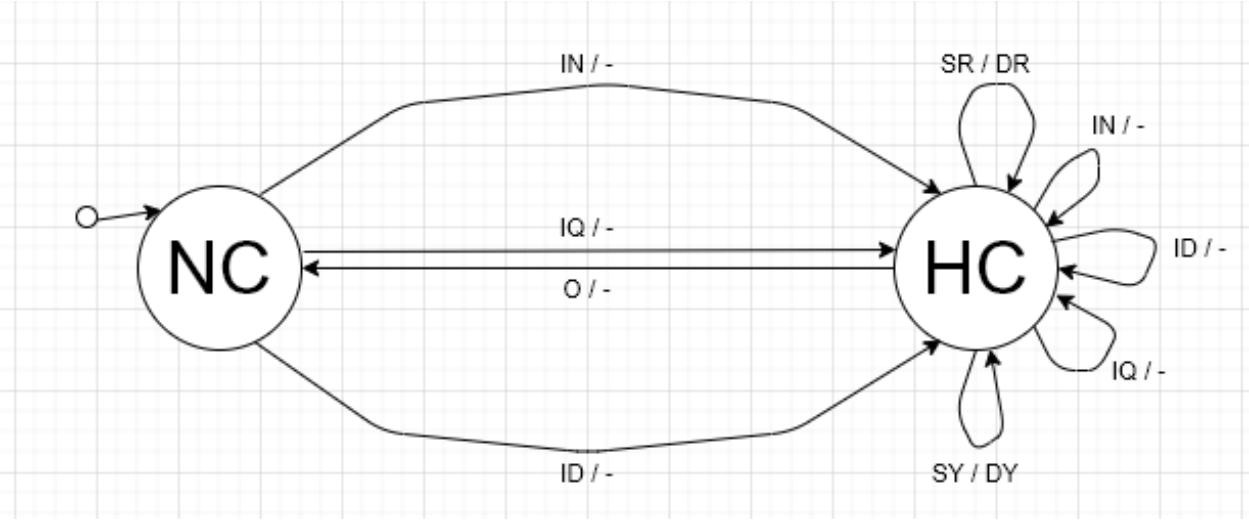


Figure 2: FSM Model of Gumball Machine (Exact Currency Subset)

| Abbreviation | Expanded Form | Meaning |
|---|---|---|
| NC | No currency | The machine has no currency |
| HC | Has currency | The machine has some currency |

Table 6: Set of States in FSM of Figure 2

| Inputs | Outputs |
|---|---|
| IN: Insert nickel | DR: Dispense red gumball |
| ID: Insert dime | DY: Dispense yellow gumball |
| IQ: Insert quarter | —: No output |
| SR: Select red gumball | |
| SY: Select yellow gumball | |
| O: Out of currency | |

Table 7: Input and Output Sets in FSM of Figure 2

| Test Case | Test Sequence | Expected Output |
|---|---|---|
| 1 | • <NC, IN / —, HC><br>• <HC, SR / DR, HC><br>• <HC, O / —, NC> | "Enjoy your red gumball" |
| 2 | • <NC, ID / —, HC><br>• <HC, SY / DY, HC><br>• <HC, O / —, NC> | "Enjoy your yellow gumball" |
| 3 | • <NC, IN / —, HC><br>• <HC, IN / —, HC><br>• <HC, SY / DY, HC><br>• <HC, O / —, NC> | "Enjoy your yellow gumball" |
| 4 | • <NC, IQ / —, HC><br>• <HC, SY / DY, HC><br>• <HC, SR / DR, HC><br>• <HC, SY / DY, HC><br>• <HC, O / —, NC> | • "Enjoy your yellow gumball"<br>• "Enjoy your red gumball"<br>• "Enjoy your yellow gumball" |

Table 8: Revised "Exact Currency" Test Cases - Conformance Testing