Homework 6

Yutong Huang (yxh589)

Problem 1

Algorithm 1: $T(\langle M, w\rangle)$

return $\langle M\rangle$

According to recursion theorem, $\exists r(w) \to \langle r(w)\rangle$

Algorithm 2: $M(w)$

Obtain $\langle M\rangle$ by invoking $T(\langle M, w\rangle)$;
Print $\langle r(\langle M\rangle)\rangle$ and halt;

$N = r(\langle M\rangle)$;
$\therefore M$ prints $\langle N\rangle$, and N prints $\langle M\rangle$

Problem 2

The following java program is submitted through canvas. A and B are implemented in such a way that they print each other.

```
/**
 * Quine - a class with two method A and B such that the class prints the source code of itself,
 *          and A and B print each other.
 */
public class Quine {

    /**
     * This method prints the start of the class, start of the main function,
     * the string that contains the partial source code, and then the rest of this method; <p>
     * Then this method invokes B() to print method A, and then invokes A() to print B; <p>
     * And finally print the close brackets.
     * @param args - no args needed
     */
    public static void main(String[] args) {
        String[] sourceOfMain = {
            "public class Quine {",
            "    public static void main(String[] args) {",
            "        String[] sourceOfMain = {",
            "            ",
            "        }",
            "        char q = 34;",
            "        for (int i = 0; i < 3; i++) {",
            "            System.out.println(sourceOfMain[i]);",
            "        }",
            "        for (String line : sourceOfMain) {",
            "            System.out.println(sourceOfMain[3] + q + line + q);",
            "        }",
            "        for (int i = 3; i < 21; i++) {",
            "            System.out.println(sourceOfMain[i]);",
            "        }",
            "        B();",
            "        A();",
            "        for (int i = 21; i < sourceOfMain.length; i++) {",
            "            System.out.println(sourceOfMain[i]);",
            "        }",
            "    }",
            "    ",
            "}"
```

```
        };

        char q = 34;
        for (int i = 0; i < 3; i++) {
            System.out.println(sourceOfMain[i]);
        }
        for (String line : sourceOfMain) {
            System.out.println(sourceOfMain[3] + q + line + q);
        }
        for (int i = 3; i < 21; i++) {
            System.out.println(sourceOfMain[i]);
        }
        B();
        A();
        for (int i = 21; i < sourceOfMain.length; i++) {
            System.out.println(sourceOfMain[i]);
        }

    }

    /**
     * Print method {@code Quine.B}
     */
    public static void A(){

        String[] sourceOfA = {
            "   public static void A(){",
            "      String sourceOfA = {",
            "          ",
            "      };",
            "      String sourceOfB = {",
            "          ",
            "      };",
            "      char q = 34;",
            "      for (int i = 0; i < 2; i++) {",
            "          System.out.println(sourceOfB[i]);",
            "      }",
            "      for (String line : sourceOfA) {",
            "          System.out.println(sourceOfB[2] + q + line + q);",
            "      }",
            "      for (int i = 3; i < 5; i++) {",
            "          System.out.println(sourceOfB[i]);",
            "      }",
            "      for (String line : sourceOfB) {",
            "          System.out.println(sourceOfB[5] + q + line + q);",
            "      }",
            "      for (int i = 6; i < sourceOfB.length; i++) {",
            "          System.out.println(sourceOfB[i]);",
            "      }",
            "   }"
        };

        String[] sourceOfB = {
            "   public static void B(){",
            "      String sourceOfA = {",
            "          ",
            "      };",
            "      String sourceOfB = {",
            "          ",
```

```
        "      };",
        "      char q = 34;",
        "      for (int i = 0; i < 2; i++) {",
        "          System.out.println(sourceOfA[i]);",
        "      }",
        "      for (String line : sourceOfA) {",
        "          System.out.println(sourceOfA[2] + q + line + q);",
        "      }",
        "      for (int i = 3; i < 5; i++) {",
        "          System.out.println(sourceOfA[i]);",
        "      }",
        "      for (String line : sourceOfB) {",
        "          System.out.println(sourceOfA[5] + q + line + q);",
        "      }",
        "      for (int i = 6; i < sourceOfB.length; i++) {",
        "          System.out.println(sourceOfA[i]);",
        "      }",
        "   }"
    };

    char q = 34;
    for (int i = 0; i < 2; i++) {
        System.out.println(sourceOfB[i]);
    }
    for (String line : sourceOfA) {
        System.out.println(sourceOfB[2] + q + line + q);
    }
    for (int i = 3; i < 5; i++) {
        System.out.println(sourceOfB[i]);
    }
    for (String line : sourceOfB) {
        System.out.println(sourceOfB[5] + q + line + q);
    }
    for (int i = 6; i < sourceOfB.length; i++) {
        System.out.println(sourceOfB[i]);
    }
}

/**
 * Print method {@code Quine.A}
 */
public static void B(){

    String[] sourceOfA = {
        "   public static void A(){",
        "      String sourceOfA = {",
        "          ",
        "      };",
        "      String sourceOfB = {",
        "          ",
        "      };",
        "      char q = 34;",
        "      for (int i = 0; i < 2; i++) {",
        "          System.out.println(sourceOfB[i]);",
        "      }",
        "      for (String line : sourceOfA) {",
        "          System.out.println(sourceOfB[2] + q + line + q);",
        "      }",
        "      for (int i = 3; i < 5; i++) {",
```

```
            "          System.out.println(sourceOfB[i]);",
            "      }",
            "      for (String line : sourceOfB) {",
            "          System.out.println(sourceOfB[5] + q + line + q);",
            "      }",
            "      for (int i = 6; i < sourceOfB.length; i++) {",
            "          System.out.println(sourceOfB[i]);",
            "      }",
            "   }"
        };

        String[] sourceOfB = {
            "   public static void B(){",
            "      String sourceOfA = {",
            "          ",
            "      };",
            "      String sourceOfB = {",
            "          ",
            "      };",
            "      char q = 34;",
            "      for (int i = 0; i < 2; i++) {",
            "          System.out.println(sourceOfA[i]);",
            "      }",
            "      for (String line : sourceOfA) {",
            "          System.out.println(sourceOfA[2] + q + line + q);",
            "      }",
            "      for (int i = 3; i < 5; i++) {",
            "          System.out.println(sourceOfA[i]);",
            "      }",
            "      for (String line : sourceOfB) {",
            "          System.out.println(sourceOfA[5] + q + line + q);",
            "      }",
            "      for (int i = 6; i < sourceOfB.length; i++) {",
            "          System.out.println(sourceOfA[i]);",
            "      }",
            "   }"
        };

        char q = 34;
        for (int i = 0; i < 2; i++) {
            System.out.println(sourceOfA[i]);
        }
        for (String line : sourceOfA) {
            System.out.println(sourceOfA[2] + q + line + q);
        }
        for (int i = 3; i < 5; i++) {
            System.out.println(sourceOfA[i]);
        }
        for (String line : sourceOfB) {
            System.out.println(sourceOfA[5] + q + line + q);
        }
        for (int i = 6; i < sourceOfB.length; i++) {
            System.out.println(sourceOfA[i]);
        }
    }
}
```

Problem 3

In class we proved that $Th(\mathbb{N}, +, \times)$ is undecidable by showing that there are some sentences in Th that are true but unprovable. This is done by constructing a Turing machine S with a hypothetical true but unprovable sentence $\varphi = \exists c[\phi_{S,0}]$ such that S only accepts if $\neg\varphi$ is proved to be true.

This fails at F_m because F_m is decidable, and thus contains no sentences that are true but unprovable.

Proof. It is obvious that Z_m is finite, therefore we can simply enumerate all the possible values to check if the sentence holds. Consider the following procedure:

1. let the sentence be: $\exists x_i[\phi_i(x_1, x_2 \dots x_i)]$;
2. iterate i from 0 to $m - 1$ and check if the sentence holds;
3. if all iterations of 2 shows the sentence holds, then the it is true, otherwise it is false.

The above procedure correctly decides F_m
$\therefore F_m$ is decidable. □