

软件质量保证 第二次作业

赵睿哲 1200012778

一、需求描述

本次作业要求实现判断一元二次方程解情况（方程形式为 $a_1x^2 + a_2x + a_3 = 0$ ，假定解在实数域）的程序，总结需求如下：

其中

$$\Delta = a_2^2 - 4a_1a_3$$
$$r_1 = \frac{-a_2 + \sqrt{\Delta}}{2a_1}$$
$$r_2 = \frac{-a_2 - \sqrt{\Delta}}{2a_1}$$
$$r_3 = \frac{-a_3}{a_2}$$

- R1: 从标准输入设备读入三个浮点数 a_1 a_2 a_3 ，输入个数或者格式错误会强制退出
- R2.1: 当 $a_1 \neq 0 \wedge \Delta > 0$ 时，输出两个以空格分割的浮点数解（ r_1 或 r_2 ）
- R2.2: 当 $a_1 \neq 0 \wedge \Delta = 0$ 时，输出一个浮点数解（ r_1 或 r_2 ）
- R2.3: 当 $a_1 \neq 0 \wedge \Delta < 0$ 时，输出 No Solution
- R3: 当 $a_1 = 0 \wedge a_2 \neq 0$ 时，输出 r_3
- R4.1: 当 $a_1 = 0 \wedge a_2 = 0 \wedge a_3 = 0$ 时，输出 Infinite（任意数值都是解）
- R4.2: 当 $a_1 = 0 \wedge a_2 = 0 \wedge a_3 \neq 0$ 时，输出 No Solution

二、代码实现

本题使用 Python 实现，核心函数为 `solve(a1,a2,a3)`，其中 `assert_params_float` 函数对输入变量的类型进行判断，这里用 `input(a1,a2,a3)`来代替并进行分析。

```
6 def solve(a1, a2, a3):
7     assert_params_float(a1,a2,a3)
8
9     if a1 != 0.0:
10         delta = a2**2 - 4*a1*a3
11         if delta < 0:
12             return "No Solution"
13         elif delta == 0:
14             return (-a2+math.sqrt(delta))/(2*a1)
15         else:
16             r1 = (-a2+math.sqrt(delta))/(2*a1)
17             r2 = (-a2-math.sqrt(delta))/(2*a1)
18             return (r1,r2)
19     else:
20         if a2 != 0.0:
21             return -a3/a2
22         else:
23             if a3 != 0.0:
24                 return "No Solution"
25             else:
26                 return "Infinite"
```

Figure 1: 代码实现

三、控制流图

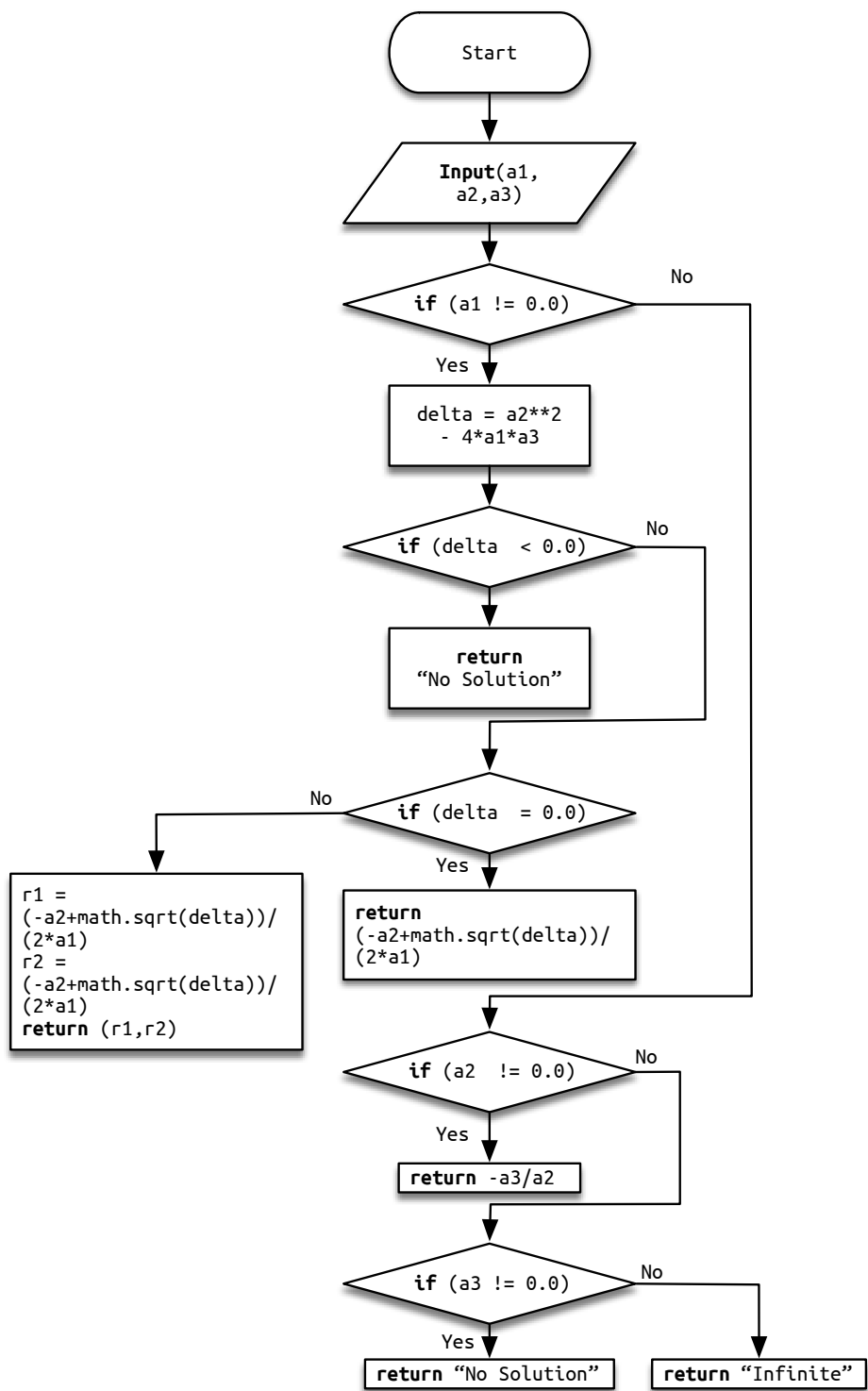


Figure 2: 控制流图

给出一组测试用例 T，其中包含：

- t1: {a1=1.0, a2=2.0, a3=0.0}
- t2: {a1=1.0, a2=2.0, a3=1.0}
- t3: {a1=1.0, a2=2.0, a3=2.0}
- t4: {a1=0.0, a2=2.0, a3=-1.0}
- t5: {a1=0.0, a2=0.0, a3=1.0}
- t6: {a1=0.0, a2=0.0, a3=0.0}

等 6 个测试用例。

语句覆盖

显然，所有测试用例输入都执行了全部的语句，因此语句覆盖充分度为 1

条件覆盖

全部简单条件有：

$a_1 \neq 0.0$: {t1,t2,t3}覆盖 true, {t4,t5,t6}覆盖 false

$a_2 \neq 0.0$: {t1,t2,t3,t4}覆盖 true, {t5,t6}覆盖 false

$a_3 \neq 0.0$: {t2,t3,t4,t5}覆盖 true, {t1,t6}覆盖 false

$\Delta > 0.0$: {t1,t4}覆盖 true, {t2,t3,t5,t6}覆盖 false

$\Delta \neq 0.0$: {t2,t6}覆盖 true, {t1,t3,t4,t5}覆盖 false

因此条件判断覆盖充分度也为 1

判定覆盖

程序中只出现了 5 个 if 判定，而且所有判定都是简单条件，因此判定覆盖充分度也为 1。

四、数据流图

Figure 3 对之前的控制流图中出现的基本块进行划分，并标记序号。每个基本块的 def, c-use, p-use 集合总结如下：

基本块	def	c-use	p-use
1	{a1,a2,a3}	{}	{a1}
2	{delta}	{a1,a2,a3}	{delta}
3	{}	{}	{}
4	{}	{}	{delta}
5	{r1,r2}	{r1,r2,a1,a2,delta}	{}
6	{}	{a1,a2,delta}	{}
7	{}	{}	{a2}
8	{}	{a2,a3}	{}
9	{}	{}	{a3}
10	{}	{}	{}
11	{}	{}	{}

数据流图如下(Figure 4)。

对 all-use 的求解：

1. 由于所有变量只定义一次，因此：

CU

$$\begin{aligned} &= dcu(a_1, 1) + dcu(a_2, 1) + dcu(a_3, 1) + dcu(\Delta, 2) + dcu(r_1, 5) + dcu(r_2, 5) \\ &= 3 + 4 + 2 + 2 + 0 + 0 \\ &= 11 \end{aligned}$$

PU

$$\begin{aligned} &= dpu(a_1, 1) + dpu(a_2, 1) + dpu(a_3, 1) + dpu(\Delta, 2) + dpu(r_1, 5) + dpu(r_2, 5) \\ &= 1 + 1 + 1 + 2 + 0 + 0 \\ &= 5 \end{aligned}$$

2. 依然使用之前的测试用例：

t1: 1-2-3, 覆盖{a1,a2,a3}的 c-use, {a1,delta}的 p-use

t2: 1-2-4-6, 覆盖{a1,a2,a3,delta}的 c-use, {a1,delta}的 p-use

t3: 1-2-4-5, 覆盖{a1,a2,a3,delta}的 c-use, {a1,delta}的 p-use

t4: 1-7-8, 覆盖{a2,a3}的 c-use, {a1,a2}的 p-use

t5: 1-7-9-10, 覆盖{a1,a2,a3}的 p-use

t6: 1-7-9-11, 覆盖{a1,a2,a3}的 p-use

a1 的所有 c-use 由{t1,t2,t3}覆盖, p-use 由{t1,t2,t3,t4,t5,t6}覆盖

a2 的所有 c-use 由{t1,t2,t3,t4}覆盖, p-use 由{t4,t5,t6}覆盖

a3 的所有 c-use 由{t1,t2,t3,t4}覆盖, p-use 由{t5,t6}覆盖

delta 的所有 c-use 由{t2,t3}覆盖, p-use 由{t1,t2,t3}覆盖

因此所有变量的 c-use 和 p-use 都被覆盖 (r1,r2 没有 c-use 与 p-use 的情况)，所以
all-use = 1.