软件测试基础与实践

实验报告

实验名称：白盒测试实验三

实验地点：计算机软件楼

实验日期：2015.11.04

学生姓名：陈建蓉

学生学号：71113106

东南大学 软件学院 制

一、实验目的

（1）巩固白盒测试知识，能熟练应用基本路径测试方法设计测试用例；
（2）学习测试用例的书写。

二、实验内容

（一）实验 1：数据流测试技术实验

运用数据流测试方法，对 CgiDecode 程序中的 decode（）方法进行测试。

实验要点：

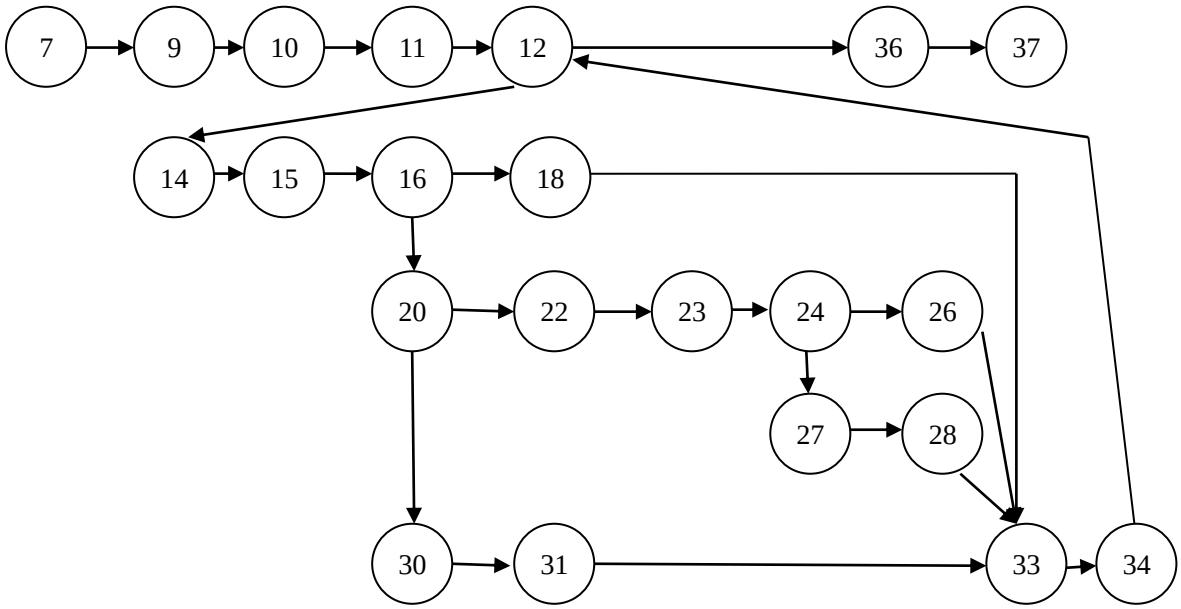
1. 约定：*eptr 和*dptr 整体作为一个变量。
由于这种变量涉及到对指针变量进行*操作，因此非声明位置出现*eptr 和*dptr 的时候都视为是相应指针变量的使用节点。

2. 难点 1：正确分析变量的定义节点和使用节点；

2. 难点 2：变量的定义节点不要求变量一定出现。
在指针发生变化的时候，会影响到相应的指针变量的值。因此，语句 22 虽然没有出现*eptr，但却是*eptr 的一个定义节点。

3. 测试过程无需运行程序，其中 getHexValue()的具体实现未给出，其作用是取对应字符串的十六进制值。

测试要考虑 decode()中 encoded, decoded, *eptr, eptr, *dptr, dptr, ok, c, digit_high, digit_low 变量：



1． 给出每个变量对应的 du-path： 代表约简后 du-path

编号	变量名	Defining nodes	Usage nodes	可能路径数	du-path
1	encoded	7	9	1	7,8,9
2	decoded	7	10	1	7,8,9,10
3.1	*eptr	9,22,23,34	12,15,22,23,31	20	9,10,11,12
3.2					9,10,11,12,13,14,15
3.3					9,10,11,12,13,14,15,16,20,21,22
3.4					9,10,11,12,13,14,15,16,20,21,22,23
3.5					9,10,11,12,13,14,15,16,20,30,31
3.6					22
3.7					22,23,24,27,28,29,33,34,12
3.8					22,23,24,27,28,29,33,34,12,13,14,15
3.9					22,23
3.10					22,23,24,27,28,29,33,34,12,13,14,15,16,20,30,31
3.11					23
3.12					23,24,27,28,29,33,34,12
3.13					23,24,27,28,29,33,34,12,13,14,15
3.14					23,24,27,28,29,33,34,12,13,14,15,16,20,21,22
3.15					23,24,27,28,29,,33,34,12,13,14,15,16,20,30,31
3.16					34,12
3.17					34,12,13,14,15
3.18					34,12,13,14,15,16,20,21,22
3.19					34,12,13,14,15,16,20,21,22,23
3.20					34,12,13,14,15,16,20,30,31
4.1	eptr	9,22,23,34	12,15,22,23,31,34	24	9,10,11,12
4.2					9,10,11,12,13,14,15
4.3					9,10,11,12,13,14,15,16,20,21,22
4.4					9,10,11,12,13,14,15,16,20,21,22,23
4.5					9,10,11,12,13,14,15,16,20,30,3

					1
4.6					9,10,11,12,13,14,15,16,20,30,31,32,33,34
4.7					22,23,24,27,28,29,33,34,12
4.8					22,23,24,27,28,29,33,34,12,13,14,15
4.9					22
4.10					22,23
4.11					22,23,24,27,28,29,33,34,12,13,14,15,16,20,30,31
4.12					22,23,24,27,28,29,33,34
4.13					23,24,27,28,29,33,34,12
4.14					23,24,27,28,29,33,34,12,13,14,15
4.15					23,24,27,28,29,33,34,12,13,14,15,16,20,21,22
4.16					23
4.17					23,24,27,28,29,33,34,12,13,14,15,16,20,30,31
4.18					23,24,27,28,29,33,34
4.19					34,12
4.20					34,12,13,14,15
4.21					34,12,13,14,15,16,20,21,22
4.22					34,12,13,14,15,16,20,21,22,23
4.23					34,12,13,14,15,16,20,30,31
4.24					34
5	*dptr	10,18,28,31,33,36		0	无
6.1	dptr	10,33	18,28,31,33,36	10	10,11,12,13,14,15,16,17,18
6.2					10,11,12,13,14,15,16,20,21,22,23,24,27,28
6.3					10,11,12,13,14,15,16,20,30,31
6.4					10,11,12,13,14,15,16,20,30,31,32,33
6.5					10,11,12,13,14,15,16,20,30,31,32,33,34,35,36
6.6					33,34,12,13,14,15,16,17,18
6.7					33,34,12,13,14,15,16,20,21,22,23,24,27,28
6.8					33,34,12,13,14,15,16,20,30,31
6.9					33
6.10					33,34,35,36
7.1	ok	11,26	37	2	11,12,13,14,15,16,20,21,30,31,

					32,33,34,35,36,37
7.2					26,33,34,36,37
8.1	c	14,15	16,20	4	14,15,16
8.2					14,15,16,20
8.3					15,16
8.4					15,16,20
9.1	digit_high	22	24,28	2	22,23,24
9.2					22,23,24,27,28
10.1	digit_low	23	24,28	2	23,24
10.2					23,24,27,28

2. 根据变量的 du-path 设计测试用例(约减后 du-path)

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
1	encoded	%61	0	a	7,8,9

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
2	decoded	%61	0	a	7,8,9,10

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
3.4	*eptr	%61	0	a	9,10,11,12,13,14,15,16,20,21,22,23
3.5	*eptr	1	0	1	9,10,11,12,13,14,15,16,20,30,31
3.10	*eptr	%411	0	A1	22,23,24,27,28,29,33,34,12,13,14,15,16,20,30,31
3.14	*eptr	%61%31	0	a1	23,24,27,28,29,33,34,12,13,14,15,16,20,21,22

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
4.4	eptr	%61	0	a	9,10,11,12,13,14,15,16,20,21,22,23
4.6	eptr	1	0	1	9,10,11,12,13,14,15,16,20,30,31,32,33,34
4.11	eptr	%411	0	A1	22,23,24,27,28,29,33,34,12,13,14,15,16,20,30,31
4.15	eptr	%63%33	0	c3	23,24,27,28,29,33,34,12,13,14,15,16,20,21,22
4.17	eptr	%411	0	A1	23,24,27,28,29,33,34,12,13,14,15,16,20,30,31
4.22	eptr	%61%31	0	a1	34,12,13,14,15,16,20,21,22,23

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
6.1	dptr	+	0	_	10,11,12,13,14,15,16,17,18
6.2	dptr	%61	0	a	10,11,12,13,14,15,16,20,21,22,23,24,27,28
6.5	dptr	1	0	1	10,11,12,13,14,15,16,20,30,31,32,33,34,35,36
6.6	dptr	%41+	0	A_	33,34,12,13,14,15,16,17,18
6.7	dptr	%61%31	0	a1	33,34,12,13,14,15,16,20,21,22,23,24,27,28

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
7.1	ok	1	0	1	11,12,13,14,15,16,20,21,30,31,32,33,34,35,36,37
7.2	ok	%%%	1	无	26,33,34,36,37

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
8.2	c	%61%31	0	a1	14,15,16,20

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
9.2	digit_high	%61%31	0	a1	22,23,24,27,28

编号	变量名	测试用例 encoded 串	输出 (ok)	decoded 串	du-path
10.2	digit_low	%61%31	0	a1	23,24,27,28

三、实验体会

1. 对于*p 和 p 的 Defining nodes 和 Usage nodes 判定容易混淆本次试验中出现了几种情况：

	*p	p
int *p;	定义	定义
p++;	定义	定义/使用
getHexValue(*(++p));	定义/使用	定义/使用
c=*p;	使用	使用
*p=" ";	定义	使用

2. 寻找 du-path 时，我们需要找到的 du-path 的数目等于定义节点数目乘以使用节点数目。
3. 通过寻找关于变量的定义和使用位置，思考程序在运行时该变量的值会如何变化。
4. 白盒测试的过程中，找出关键有效的测试用例是关键，找到关键的测试用例可以有效控制测试用例数目。